

MINISTRY OF EDUCATION AND TRAINING



Veazy Team			
Group Members	Nguyễn Ngọc Minh	SE03174	Minhnse03174
	Nguyễn Trọng Duy	SE03235	Duyntse03235
	Nguyễn Hoàng Linh	SE03295	Linhnhse03295
	Đào Thanh Tùng	SE03056	Tungdtse03056
	Phạm Đức Thắng	SE02722	Thangpdse02722
	Nguyễn Hồng Quân	SE02920	Quannhse02920
Supervisor	Mr. Nguyễn Văn Sang	Lecturer at FPT University	
Capstone Project code	Veazy		

Hanoi, 24th August, 2016

• Table of Contents

Definition and Acronyms	5
1 Introduction	7
1.1 Project information	7
1.2 Project members.....	7
1.2.1 Supervisors	7
1.2.2 Team members	7
1.3 Purposes	8
1.4 Abstract.....	8
1.5 Literature Review.....	9
1.6 Proposal.....	11
1.6.1 The idea	11
1.6.2 Objectives	11
1.6.3 Brief description about system	11
1.6.4 System features.....	11
1.7 Benefit From Project.....	12
1.7.1 For development team	12
1.7.2 For Community.....	13
1.8 Critical Assumption and Constraints	13
1.8.1 Assumption.....	13
1.8.2 Constraints	13
1.9 Potential Risk	14
2 Software Project Management.....	15
2.1 Project Overview	15
2.1.1 Project Description	15
2.1.2 Purposes of Project	15
2.1.3 Scope Statement	15
2.1.4 The functions of Project	16
2.2 Project Organization	18
2.2.1 Organizational Structure.....	18
2.2.2 Project Process.....	19
2.3 Project Schedule.....	21
2.3.1 Milestone and Deliverables	21

2.3.2	Gantt chart	23
2.4	Risk Management	24
2.5	Communication Management.....	25
2.6	Tools and Infrastructure	28
2.7	Configuration Management Plan	28
3	Software Requirement Specification	29
3.1	Overview.....	29
3.1.1	Purpose	29
3.1.2	Scope	29
3.2	User Requirements Specification.....	29
3.2.1	Business Process Overview	29
3.2.2	Product Features	40
3.3	Software Requirements Specification.....	41
3.3.1	Overall Description.....	41
3.3.2	Entity Relation Model	45
3.3.3	Functionality.....	54
3.3.4	Non-Functional Specification Requirements.....	117
4	Software Design Description.....	119
4.1	SDD Overview.....	119
4.2	Architecture Design	119
4.2.1	MVC Model.....	119
4.2.2	SpringMVC framework.....	121
4.2.3	HibernateORM framework.....	124
4.2.4	AngularJS framework.....	126
4.2.5	Architectural Representation	129
4.2.6	Architecture Goals and Constraints	130
4.2.7	Use-Case View	131
4.2.8	Logical View	140
4.2.9	Process View	147
4.2.10	Deployment view.....	154
4.2.11	Class Design	155
4.3	Screen Design	208
4.3.1	Screen Flow	208

4.3.2	Guest.....	210
4.3.3	Logged-in user.....	222
4.3.4	Content creator	229
4.3.5	Admin	251

DEFINITION AND ACRONYMS

Acronym	Definition	Note
VEAZY	Vietnamese Study System for Japanese	
SRS	Software Requirements Specification	
IT	Information Technology	
JS	JavaScript	
FU	FPT University	
CP	Capstone Project	
SQL	Structured Query Language	
ADD	Architecture Design Document	
CI	Configuration Item	
CM	Configuration Management	
PP	Project Plan	
CSCI	Computer Software Configuration Items	
DDD	Detail Design Document	
PM	Project Manager	
QA	Quality Assurance Officer	
Source	Source Code	
URD	User Requirement Document	
TP	Test Plan	
PIC	Person in Charge	
TL	Test Leader	
TP	Test Plan	
TC	Test Case	
PTL	Project Technical Leader	

Acknowledgements

During the implementation of this project, we have received overwhelming support from a number of people, to whom we would like to express our great appreciation. Without them, this project would have been done with much more obstacles than it was.

First and foremost, we would like to show our thankfulness to Lecturer Nguyễn Văn Sang, our supervisor of this project during four months and teacher Do Thi Van, specialized consultants, who taught us Japanese. They have given us not only many valuable advices but also enthusiastic inspirations that helped raising our team members' spirit to give us the possibility to complete this Capstone Project.

Last but not least, we are thankful to our families, our teachers and our friends in FPT University for their constant encouragement and support throughout this project.

1 INTRODUCTION

1.1 Project information

- **Project Name:** Vietnamese Study System for Japanese
- **Project Code:** VEAZY
- **Project Type:** Web application
- **Duration:** From 9th May 2016 to 1st September 2016

1.2 Project members

1.2.1 Supervisors

Full name	Phone	E-Mail	Position
Nguyễn Văn Sang	0912210727	sangnv@fpt.edu.vn	Supervisor

1.2.2 Team members

Full name	Roll No.	Phone	E-mail	Position
Nguyễn Ngọc Minh	SE03174	01645698393	mr.noone1945@gmail.com	PM + PTL
Nguyễn Hồng Quân	SE02920	0963236323	quannh265@gmail.com	Test Leader
Phạm Đức Thắng	SE02722	01642679718	thangpdse02722@fpt.edu.vn	Developer
Nguyễn Hoàng Linh	SE03295	01655108408	linhnhse03295@fpt.edu.vn	Developer
Nguyễn Trọng Duy	SE03235	0975572524	fujin0904@gmail.com	Developer
Đào Thanh Tùng	SE03056	01687559224	trumdongnat@gmail.com	Developer

Table 1-1: Veazy • team members

1.3 Purposes

This document is created as the introduction for project Vietnamese Study System for Japanese (Veazy). The purpose of this document is to convey information about:

- (a) Background overview & literature review of similar existing system.
- (b) Brief introduction about the initial idea for Veazy.
- (c) Main features of Veazy.
- (d) Constraints and potential risks of the project.

1.4 Abstract

Vietnam is ranked as one of the top 10 nations that attract foreign labor candidates. In order to adapt to new living and working environment, a lot of foreign people who stay and work in Vietnam have high need in studying Vietnamese as a second language.

However, because Vietnam is not seen by media as economic powerhouse, thus, in comparison with English or Japanese, Vietnamese is far from becoming a common language all over the world. Along with its complexity and unclear curriculum structure, Vietnamese causes many difficult to learners, especially those who struggle learning by themselves. Besides, due to working schedules, many people find it difficult to afford their time to enroll a Vietnamese course at the language centers.

Aiming at foreign people working in Vietnam, in particular, Japanese people and those who have interest in Vietnamese as well and striving to achieve the objective of helping users study Vietnamese in an easy and convenient way, the website is expected to solve all limitation above.

1.5 Literature Review

Although Vietnamese is not ranked as one of the most common languages, there still are websites and mobile applications that support users in studying Vietnamese. Each of them has their own unique traits as well as advantages and disadvantages. These following are some typical examples of existing systems which team members carried out analysis:

123vietnamese.com

The screenshot shows the homepage of 123VIETNAMESE. At the top, there's a navigation bar with links: HOME, ON-SITE COURSES, LINKS, LEARN ONLINE, BLOG, and CONTACT. Below the navigation is a large banner featuring a woman smiling and the text "VIETNAMESE COURSE FOR BEGINNERS". Underneath the banner, it says "We provide Vietnamese classes at our school or at your home/ office also." The main content area is titled "\$ Price" and contains four colored boxes representing different course options:

Price	Duration	Class Type
\$11.6 \$9.3 /hour	63 hours	One on One
\$14.4 \$11.5 /hour	63 hours	One on One
\$5.4 /hour/student	63 hours	Group Class
\$6.3 /hour/student	63 hours	Group Class

Figure 1. 123Vietnamese

123VIETNAMESE is a Vietnamese language training center that was founded in 2010 by a teacher group from Hanoi University of Education and University of Languages & International Studies.

- Advantages:
 - The website supports user in studying through video call with lecture, which is flexible in time for students.
 - The diverse online lessons are classified based on level of difficulty and purposes (for academy, research or travelling).
- Disadvantages:
 - The website charges course fee.

I-lingo.com

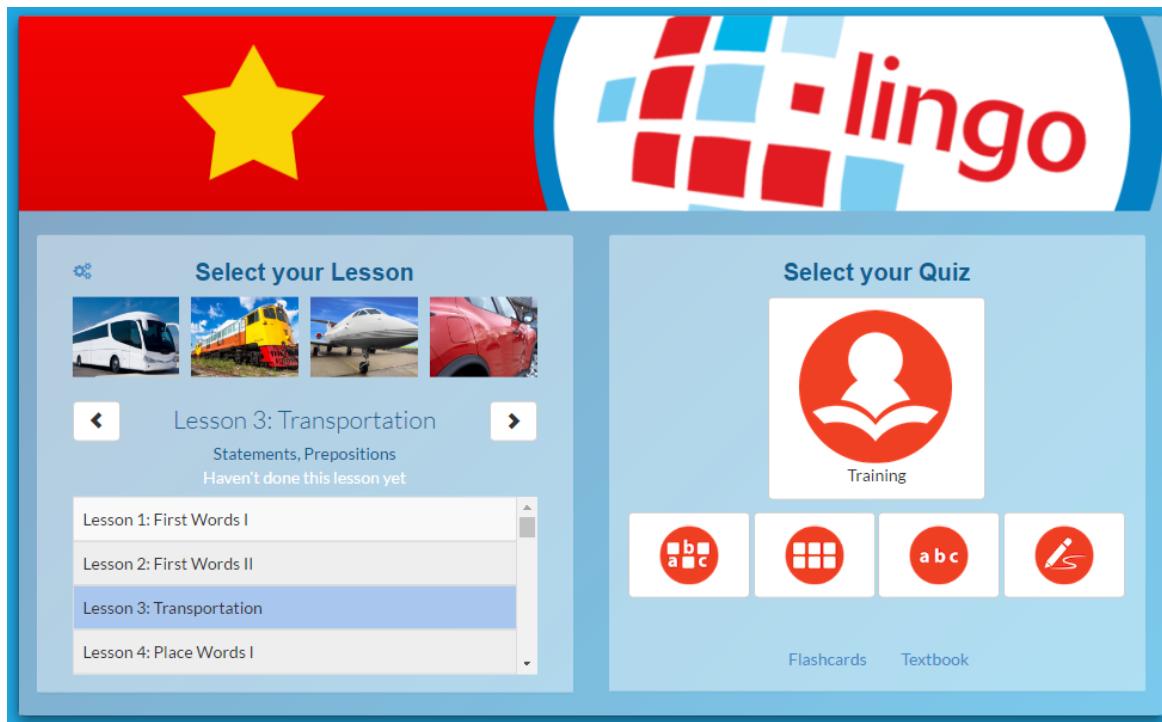


Figure 2. I-lingo

I-lingo is a website helping users study 20 Asian & European languages including Vietnamese.

- Advantages:
 - The themes of lessons are very diverse.
 - The website uses words, images and sound clips of native speakers, which is very convenient for users to study vocabulary and practice listening.
 - The website including quiz section at each lesson to help user review.
- Disadvantages:
 - The website focuses mostly on teaching vocabulary related to themes. There is no grammar section in each lesson.
 - The number of free lessons is limited. User have to register and log in in order to continue studying.

1.6 Proposal

1.6.1 The idea

Currently, there are lots of applications supporting foreign people to study Vietnamese which diversifies from websites to mobile applications. Still, after having analyzed carefully some of existing systems, team members have found out several disadvantages listed above.

The idea of Veazy is to overcome those limitations by creating a free website for studying Vietnamese, in which our team keeps original basic features such as lessons, small tests while including convenient features for users as well as features adapting to changes in study course's structure.

1.6.2 Objectives

Capstone Project (CP) is the final subject included in study program of FPT University (FU). There are 4 main objectives of a CP.

The first and foremost objective of this project is to *achieve the objective which project team set at the beginning*: to release a website supporting users in study Vietnamese. The second object is that this CP serves as a *standard* for CP Council to *decide whether development team is qualified to graduate or not*. The third objective of CP is to help students *review all knowledge related to IT technique and IT project management* studied through 4 years in university. During project management process, students learn how to apply these knowledge and make decision in order to lead the project to success. The fourth objective is to help students *acquire self-study skill through making research* about programming languages and popular frameworks for the need of projects which are not included as a part of the university study program.

1.6.3 Brief description about system

This software is a web application which offers users almost of main features without registration. The website help users enlarge knowledge about Vietnamese including grammar, vocabulary through activities such as studying by lesson, reading articles and looking up dictionary, taking test corresponding to each skill and level of difficulty. User can send feedback to admin about the contents of the website. After logging in the system, user can see their progress of study and skills assessments.

Veazy uses familiar technologies such as Java, JavaScript, HTML, PostgreSQL and supports the most used browser like Internet Explorer, Firefox or Chrome.

1.6.4 System features

Veazy's features are divided into 4 different categories based on types of users:

- **Features for ordinary users:**
 - **Study Vietnamese:** Users can select and enroll in an appropriate course of Vietnamese divided into 6 levels of difficulty. Each lesson contains 6 sections of content: vocabulary, grammar, conversation, listening, practice and reading.
 - **Take Vietnamese tests:** Users can take multiple choice test based on difficulty level and type of skill which user can select.
 - **Report content:** User can send report about the contents of lessons and test questions.

- ***Look up in dictionary:*** Users can search for words' meanings and related examples in both Vietnamese-Japanese and Japanese-Vietnamese dictionary.
- ***View statistics (only available after login):*** Users can display charts representing current progress of study, assessment on each respective skill.
- ***Log in/Log out:*** Users can log into Veazy system with a registered account for more unique futures. Users can register new account or choose to log in by using Facebook or Google account.
- ***Change password/Recover password:*** Users can change password when necessary or request to update a new one in case of forgetting the current password.
- **Extra features for content editors:**
 - ***Review report:*** Admins & content creators can check the reports or feedbacks sent by users.
 - ***Manage lesson contents:*** Content creators can view list of lessons in each course, create, edit content and delete a lesson.
 - ***Manage test question contents:*** Content creators can view list of questions, create, edit content and delete a question.
- **Extra features for admins:**
 - ***Manage users:*** Admins can view list of accounts as well as each respective account's detail information, assign authority (admin or content creator) to any account, ban/unban an account.

1.7 Benefit From Project

1.7.1 For development team

All members in development team gain more experience in each following field:

- **IT Project management:** Team members understand all phases of project management as well as individual responsibility, what needs to achieve, what needs to deliver during an IT project.
- **Programming languages:** Front-end developers gain more experience in HTML&CSS, JavaScript while back-end developers gains more experience in Java and PostgreSQL.
- **Soft skills:** Each member learns to become a team player who can communicate effectively and know how to delegate and assign tasks to appropriate members as well as support each other.
- **Natural languages:** Team members have the chance to make research about Vietnamese and Japanese at a certain level of depth, which enrich members' knowledge. Members also get to realize and correct inaccurate ways of speaking and writing Vietnamese which are often mistaken.

1.7.2 For Community

All Internet users have more resources and channel which are free of charge and allow:

- Studying Vietnamese grammar and vocabulary through online courses.
- Keeping track of study progress and current ability.
- Enhancing reading ability as well as knowledge through reading articles about Vietnam.
- Taking test corresponding to level of difficulty.
- Taking advantage of Vietnamese-Japanese dictionary during learning process.

1.8 Critical Assumption and Constraints

1.8.1 Assumption

An assumption is a circumstance or event that need to occur for the project to be successful, but are out of range project team can control. These following are circumstances assumed to occur in project:

- Devices used to store data of project are well preserved.
- When a team member becomes sick, there is another members who can take over and continue his or her tasks without delaying project progress.
- Team members are able to use new technologies need for the project after 3 weeks of training.
- Team members are able to devote full 6 hours per day for project commitment.
- Team members can communicate well with each other.

1.8.2 Constraints

A constraint, in project management, is any restriction that defines a project's limitation. These following are project constraints defined by team members and classified into categories:

- **Time:** The length of project is 4 months (from May to August, 2016). Project must be delivered and ready for demo before the due date of August 25th.
- **Resources:**
 - Team scales is limited to 6 people who are all junior software developer.
 - Team members spend at least 6 hours per day for the project.
- **Quality:**
 - The website should operate well on popular web browsers such Firefox (version 40.0 above) and Google Chrome (version 45.0 above).
 - The website should provide full features listed at part 4.4.

1.9 Potential Risk

Project risk is an uncertain event or condition which has an effect on project objective if it occurs. These following are project risks identified which might have the possibility to happen:

- **Insufficient PM Experience:** Schedules do not meet because of shortage of PM experience. PM needs to prepare knowledge and learn from other PMs' experience and supervisor.
- **Too many requirement changes:** Initiative requirement maybe not suitable or clear. It is be changed while the project is running.
- **Shortage of Human resource:** Member leaves the project team or fall seriously ill.
- **Not meeting performance requirements:** The lack of experience causes low quality Japanese documents.
- **Unclear requirements:** Member don't understand the requirements.
- **Lack member experience:** Member cannot afford requirements, low quality work
- **Member conflict:** Conflicts between members caused by misunderstanding during communication or being unable to reach a consensus can affect teamwork, which decreases productivity.

2 SOFTWARE PROJECT MANAGEMENT

2.1 Project Overview

2.1.1 Project Description

Project Code	Veazy	Contract Type	None
Customer	FPT University	2nd Customer	None
Project Level	Group	Project Rank	None
Application Type	Website	Project Manager	Nguyen Ngoc Minh
Project Category	Development	Business Domain	Education

Table 2-1: Project description

2.1.2 Purposes of Project

This project is a capstone project of our group at FPT University. By releasing this project, we want to not only pass the capstone challenge but build a website that helps Japanese people, who are interested in Vietnam, study about Vietnam as well. After analyzing the business specifications, we found out that there is too few useful resources which can help foreigners, especially Japanese people, study about Vietnamese language and culture. Even both Japanese and Vietnamese are affected by Chinese culture, but it is still quite difficult to understand each other. Thus, our product is on purpose to provide a system to help Japanese people study about Vietnam in a way that is as natural as possible.

On the other hand, through 4 months of this project, we hope that every member can gain more experience and knowledge about software development, improve personal and group working skills.

2.1.3 Scope Statement

The scope of this project contains: Requirement Analysis, Design, Coding and Testing (Unit Test, Integration Test and System Test).

2.1.4 The functions of Project

- **User account:** Allows users register new accounts in order to log into the system and provide features of updating and viewing profiles.
- **E-learning system:** Provides a well-organized lessons system divided based on level of difficulty.
- **Testing system:** Allow users to take after setting up as well as viewing the test result.

No	Actor	Use-case name	Description
1	Guest	Register	Registering new account to log into the system
2		Log In	Logging into the system
3	Guest, Member	View Lesson List	Viewing lesson list of courses
4		View Lesson Details	Viewing detail of lesson
5		Create Test	Creating a test
6		Take Test	Taking a multiple-choice test
7		View Test Result	Viewing result of test
8		Change Language	Changing language
9	Member	View Profile	Viewing detail information of their profile
10		Update Profile	Updating user's profile
11		Change Password	Changing user's password
12		View Statistics	Viewing statistics of user
13		Log Out	Logging out of system
14		View List Test History	Viewing list of test history
15		View Test History	Viewing detail of test history

16		Retake Test	Retaking a test in list test history
17		Report Lesson	Reporting content of lessons that having wrong content
18		Report Question	Reporting content of questions that having wrong content
19	Content Editor	View Profile	Viewing detail information of their profile
20		Update Profile	Updating user's profile
21		Change Password	Changing user's password
22		Log Out	Logging out of system
23		View Lesson List	Viewing lesson list of courses
24		Filter Lesson	filtering lesson
25		View Lesson Details	View detail of lesson
26		Edit Lesson	Editing content of lesson
27		Delete Lesson	Deleting lesson
28		Create Lesson	Creating new lesson of course
29		View Questions List	Viewing question list
30		Filter Question	Filtering question
31		View Question Details	View detail of question
32		Edit Question	Editing content of question
33		Delete Question	Deleting question
34		Create Question	Creating new question of course
35	Administrator	View Profile	Viewing detail information of their profile
36		Update Profile	Updating profile

37	Change Password	Changing password
38	Log Out	Logging out of system
39	View Users List	Viewing list of user
40	Search User	Searching user
41	View User Profile	Viewing detail information of user
42	Ban User	Blocking user of system
43	Unban User	Unblocking user
44	Set User Role	Setting role for user
45	View System Statistics	Viewing statistics of system

Table 2-2: List function

2.2 Project Organization

2.2.1 Organizational Structure

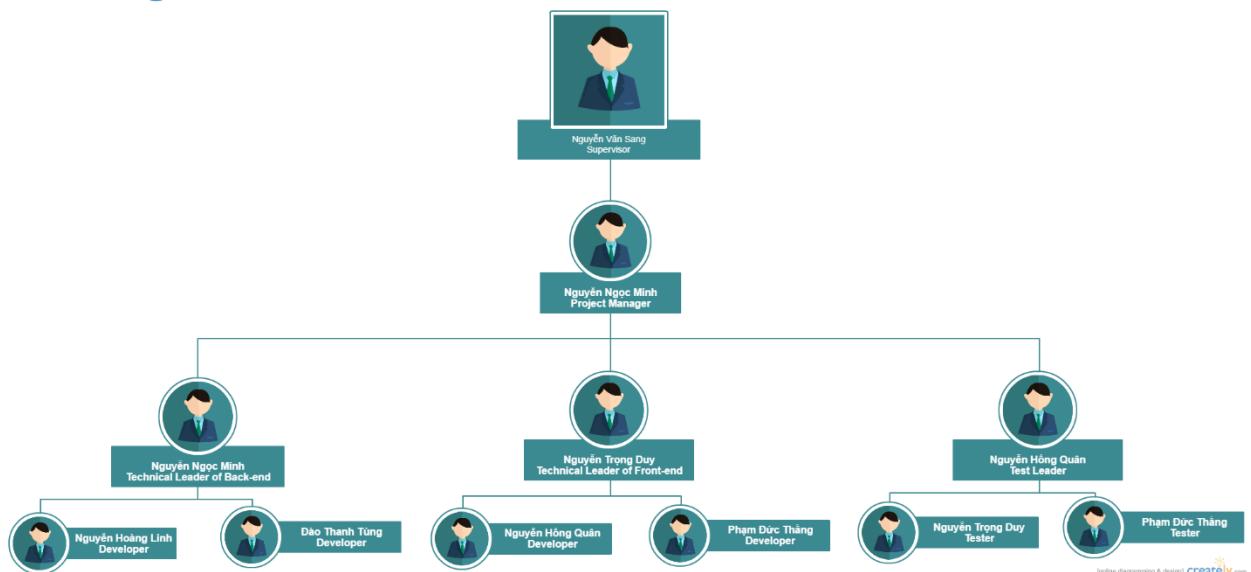


Figure 2-1 Project organizational structure

2.2.2 Project Process

2.2.2.1 Fsoft Process Model

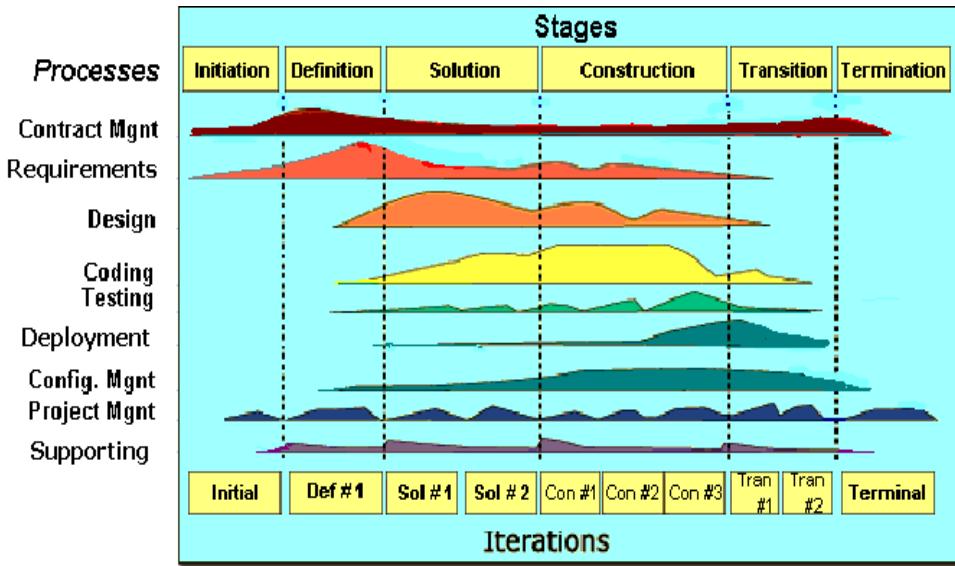


Figure 2-2 Fsoft Process Model

The software lifecycle is broken into *cycles*, each cycle working on a new generation of the product. The Fsoft process divides one development cycle in six consecutive *phases*:

- ✓ Initiation phase
- ✓ Definition phase
- ✓ Solution phase
- ✓ Construction phase
- ✓ Transition
- ✓ Termination

2.2.2.2 Project Life Cycle

Basing on Fsoft process and real-world project, we decided to divide the project into four phases: Initiation, Solution, Construction, and Termination:

- **Initiation Phase:** This is the explanatory phase of the project. Project objective and description is described at this stage. The purpose of this phase is to collect and understand business requirements, detail the project plan and agree upon a high level statement of work. Our primary objectives are complete project identification and project plan. After these are completed, the project is checked against the following criteria:
 - Identify business functions of the system
 - Determining the scope, conditions and limitations of the project
 - List the main functions of the system
 - List one or more suitable architecture for the system
 - Identify project risks

- Complete Report 1, and Report 2

- **Solution Phase:**

- In this phase, the architecture of the system is designed. The goal is to translate requirements and specification into a technical solution to produce Technical Design. Our primary objectives are complete requirement specification, architecture design and database design.
- Finally, the plan must provide (including estimates of cost and time) for the construction phase. The plan must ensure proper and accurate based on experience.
- Complete Report 3 and Report 4

- **Construction Phase:**

- This is the longest phase of a project life cycle.
- In this phase, all functions of the system will be installed. The installation will be divided into small stages, each stage of the installation a few functions. The results of each phase will be the release of the module function can be executed.
- Construction and improvement of products until the final product is ready to deliver to the user. During this phase, all the components and other features of the application is developed and integrated into the product.
- This phase emphasizes the resource management and control operations to optimize cost, time and quality.
- Complete software packages and Report 5, Report 6

- **Termination Phase:**

- This is the final phase in the life cycle of a project.
- Their products will be deployed to the client. The feedback received during the transfer process will be recorded and put on the new functional requirements or functionality enhancements in the next version of the product.
- Phase transfer switch also includes the training system and the new system for the user.

2.3 Project Schedule

The Effort estimation is documented in the table below.

	Initiation	Solution	Construction	Termination	Total
Effort(person/day)	75	147	198	30	450
Total % budgeted Effort Usage (%)	100	100	100	100	

Table 2-3: Effort Estimate

2.3.1 Milestone and Deliverables

No	Stage	Committed Delivery date	Description of Deliverable	Delivery media
Initiation			Requirements agreed, Report 1 reviewed	
1	Project Introduction	16-05-2016	Criteria: Documentation reviewed	
2	User Requirement Specification	16-05-2016	Criteria: Documentation reviewed	
3	Screen mockups	19-05-2016	Criteria: Documentation reviewed	
4	Project Plan	24-05-2016	Criteria: Documentation reviewed	
Solution				
1	Architecture Design	27-05-2016	Criteria: Documentation reviewed	
2	SRS	01-06-2016	Criteria: Documentation reviewed	
3	Detail Data Design	06-06-2016	Criteria: Documentation reviewed	
4	Screen Design	07-06-2016	Criteria: Documentation reviewed	
5	Class Design	07-06-2016	Criteria: Documentation reviewed	
6	Test Plan	09-06-2016	Criteria: Documentation reviewed	

7	Integration Test Case	29-06-2016	Criteria: Documentation reviewed	
Construction			Product developed & tested and released to supervisor, documentation reviewed.	
1	Source code	01-08-2016	Source code Acceptance criteria: Product unit tested	
2	System Test Case	07-07-2016	Criteria: Documentation reviewed	
3	Test report	12-08-2016	Criteria: Documentation reviewed	
4	Final report document and CD source code	24-08-2016	Final Documents and Source Code	
Termination			Project post-mortem is conducted, Project assets archived and released to supervisor	
1	Lesson learned	01-09-2016	Criteria: Completed	

Table 2-4: Milestone and Deliverables

2.3.2 Gantt chart

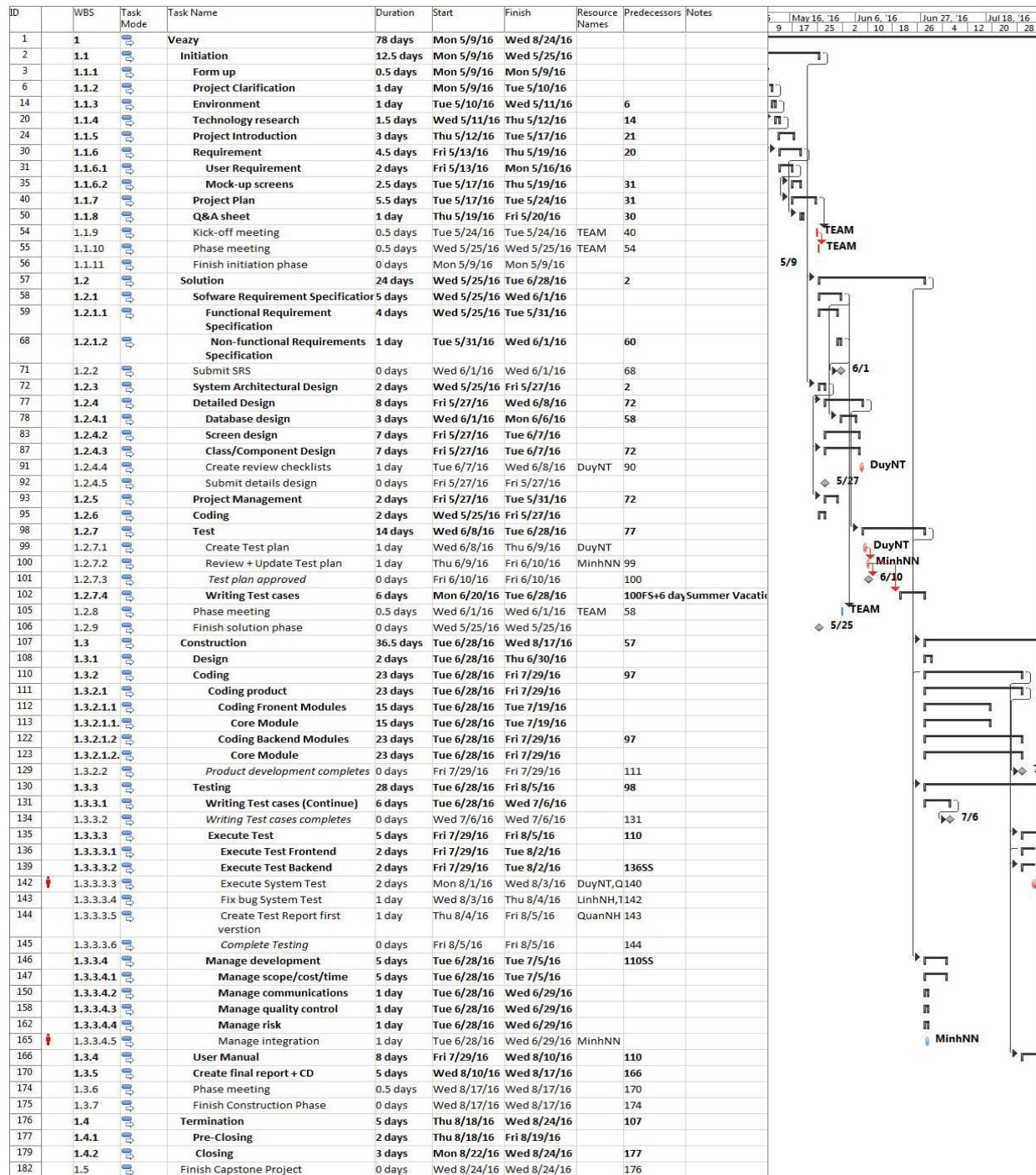


Figure 2-3: VEAZY Gantt chart

2.4 Risk Management

RISK ASSESSMENT MATRIX

Risk ID	Risk Category	Risk Source	Risk Description	Probability	Impact	Exposure	Priority	Trigger	Risk Status	Last updated date	Mitigation	Contingency	Person in charge	Planned end date	Mitigation Cost	Mitigation Benefit	Actual end date
R01	Human resource	Insufficient PM Experience	Unrealistic schedules	0.8	3	6.4	8	Deliverables are over 50% late when meet the deadlines	Open		PM needs to prepare knowledge and learn from other PMs' experience	PM needs to ask support from the supervisor.	PM	8/25/2016			8/25/2016
R02	Project Requirement	Too many requirement changes	SRS has been changed too frequently	0.3	6	1.8	6	SRS has been changed over 10 times a week after SRS was approved.	Open		Clear out the requirements and members' abilities at the beginning of the project	Pending all change requests, have meeting to clear the requirements again	PM	8/24/2016			8/29/2016
R03	Human resource	Shortage of Human resource	Lack of team members	0.3	4	1.2	4	Member leaves the project team	Open		Have an open talk with team member to clear out the objectives and understand each others.	Assign work to other members; increase effort	PM	8/24/2016			8/25/2016
R04	Human resource	Not meeting performance requirements	Low quality Japanese documents	0.3	4	1.2	4	Low quality Japanese documents (too many mistakes)	Open		Have plans to train Japanese for team members	Request support from Japanese lecturer	Team members	8/25/2016			8/25/2016
R05	Project Requirement	Unclear requirements	SRS is not clear enough	0.4	4	1.6	4	Member cannot understand the requirements	Open		Define clearly the requirements	Have meetings to define and clear the requirements again	PM	8/24/2016			8/25/2016
R06	Human resource	Team Spirit and Attitude	Member conflict	0.4	4	1.6	4	Member can be misunderstand each other	Open		Hold team building to increase members' relationship	conciliate the conflict	PM	8/24/2016			8/27/2016
R07	Human resource	Not meeting performance requirements	Lack of members' experience	0.3	3	0.9	3	Member cannot afford requirements, low quality work	Open		Train members and run a pilot project before kicking off the main project	Break works, share efforts between lower experienced member and the higher one.	Team members	8/26/2016			8/26/2016

2.5 Communication Management

Communication Type	Method / Tool	When	Information	Participants / Responsible
Project Task Tracking				
Task scheduling	MS Project Professional 2010	At the beginning of every stage, and weekly Refinement and rescheduling as necessary		PM
Task assignment	MS Project Professional 2010	Weekly		PTL
Task status reporting	Daily Report	Daily		Project Team Members
Project Meeting				
Kick-off Meeting	Face to face	Initiation stage	Project introduction; Project plan review; Risk identification; stakeholders identify.	PM, Project Team Members
Project Progress Review Meetings	Face to face	Weekly and on event	Communicate project status Communicate and resolve any open issue, risks, and changes Discuss any suggested improvement	PM, Project Team Members

Communication Type	Method / Tool	When	Information	Participants / Responsible
Milestone Meetings	Face to face	1 days After the completion of stages: Definition, Solution & Construction	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	PM, Project Team Members, QA, Supervisor
Transfer/Sharing of project documentation/in formation	Github	When available	All project documentation and information	PM, Project Team Members, QA
Supervisor Communication and Reporting:				
Project Report	Agreed Fsoft and FU standard format	5pm Monday, Weekly	Project status report, Issue requiring clarifications, escalation, if any	PM
Project Meetings with supervisor	Face to face	12h45 Friday, Weekly	As above	PM
Requirement gathering/clarification	Face to face meeting	During requirement analysis phase	As in Q&A list	PM
Communication with Supervisor				
Review Project Plan & Project schedule	By attend project meeting	Significant changes to WO, PP and Project schedule (scope, objectives Organization, HR, major milestone, deliverables)		PM

Communication Type	Method / Tool	When	Information	Participants / Responsible
Project Progress Review	By email and/or via Operation meeting at Group/Division level	Weekly	Project status report, Issue requiring clarifications, escalation, if any	PM
Project Milestone Review	By email and via project milestone review meeting	End of every stage	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	PM

Table 2-6: Communication Management

2.6 Tools and Infrastructure

Below is the list of tools and infrastructure requirements needed for development environment.

Item	Description	Expected Availability by	Note
Development Environment			
Operating System	Window 7 or above version, Ubuntu LTS 14.04, CentOS 6.6		
Browser	Google Chrome, Firefox (all version)		
Development language	Java 8		
Technology			
Development language	Java 8		
Server	Tomcat 7 (server localhost)		
Hardware Requirement			
Hardware Configuration	2GB workspaces on server		
Equipment & Tools			
Source Version Control	Git	Definition stage	
Task Tracking	MS Project Professional 2010	Initiation stage	
SRS	Microsoft Office Word, Microsoft Office Excel, Astah Professional	Initiation stage	

Table 2-7: Tools and Infrastructure

2.7 Configuration Management Plan

The detail configuration management is available in file [Veazy_CMPlan_v1.0_EN.docx](#).

3 SOFTWARE REQUIREMENT SPECIFICATION

3.1 Overview

3.1.1 Purpose

The purpose of this document is to present a detailed description of the Veazy. It explains the interface and the features of the system, what the system does, the constraints under which it must operate and how the system reacts to external stimuli...

3.1.2 Scope

Veazy is a website which supports Japanese people study Vietnamese.

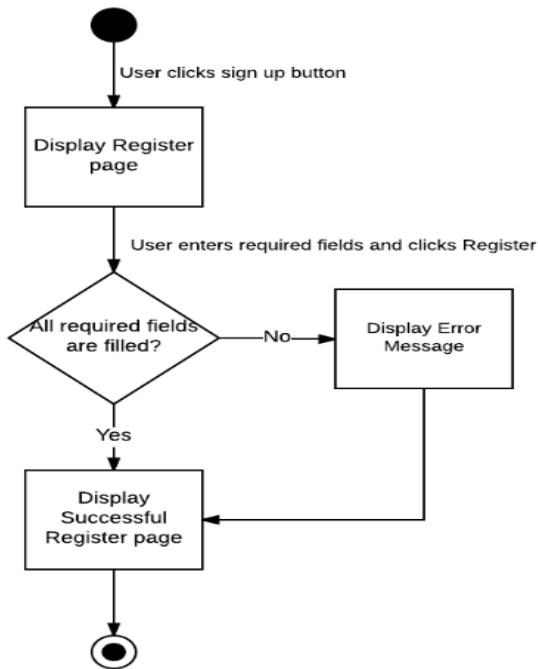
There are four basic users Guest, Member, Content Editor and Admin.

- Guest can study via lessons and do test.
- Member can access basic functions of guests and manage profile, redo recently taken tests and view statistics assessed based on those taken tests.
- Editor can manage website content which contains of contents of lesson and test question.
- Administrator can manage users, including checking user detailed information, assign authority and ban/unban.

3.2 User Requirements Specification

3.2.1 Business Process Overview

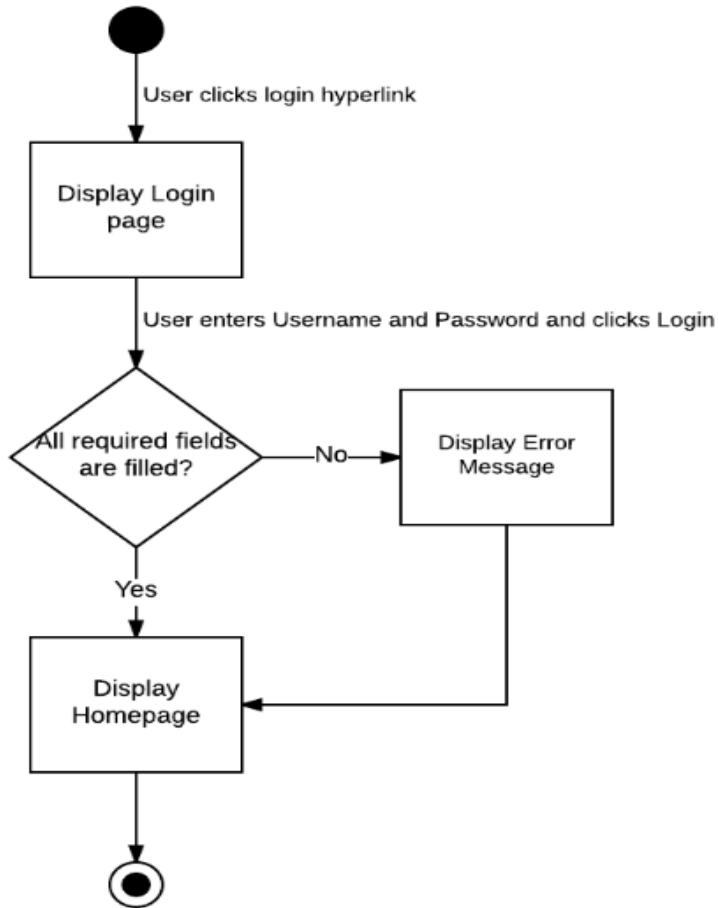
3.2.1.1 Register workflow



Brief description: Guest can use this function. Steps:

- Step 1: User clicks on Sign Up button.
- Step 2: System display Sign Up page.
- Step 3: User enters required fields and clicks on Register button.
- Step 4: System display Successful Register page or Error Message.

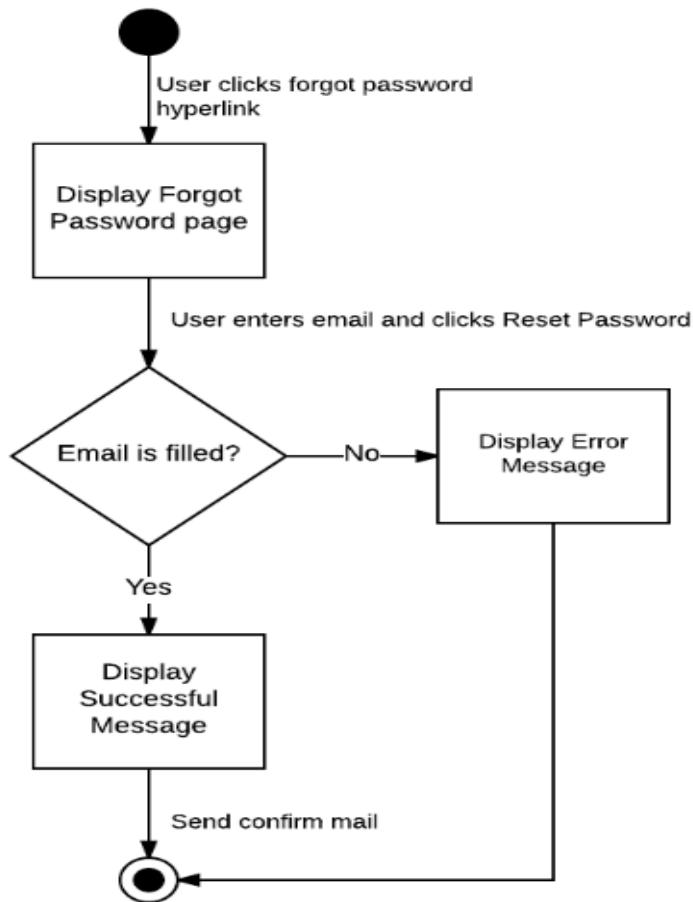
3.2.1.2 Login workflow



Brief description: Guest can use this function. Steps:

- Step 1: User clicks on Login hyperlink.
- Step 2: System display Login page.
- Step 3: User enters Username and Password and clicks on Login button.
- Step 4: System display Homepage or Error Message.

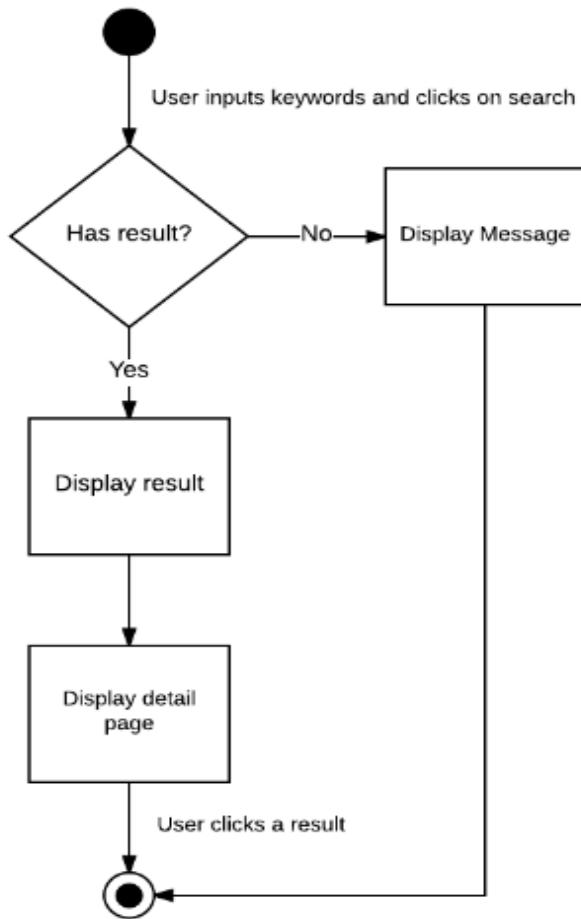
3.2.1.3 Forgot Password workflow



Brief description: Guest can use this function. Steps:

- Step 1: User clicks on Forgot Password hyperlink.
- Step 2: System display Forgot Password popup.
- Step 3: User enters email and clicks on Reset Password button.
- Step 4: System display Successful Message or Error Message.

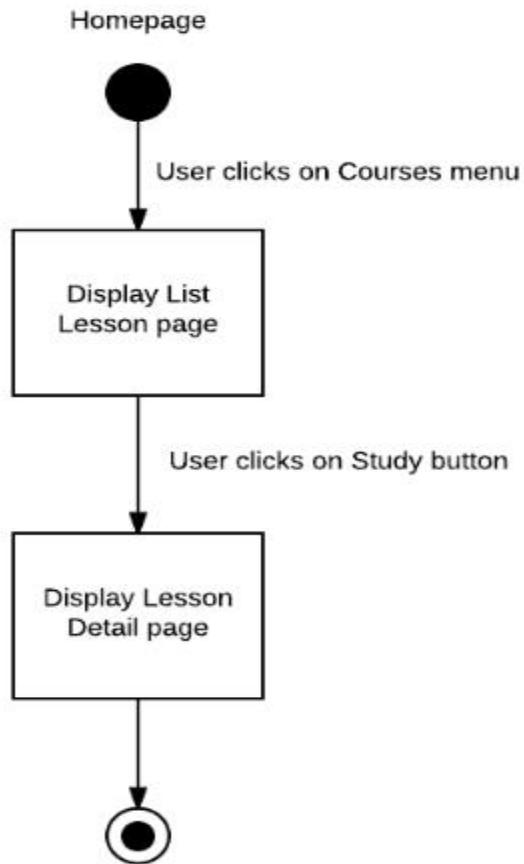
3.2.1.4 Search workflow



Brief description: Guest and Member can use this function. Steps:

- Step 1: User inputs keywords into search textbox and clicks on Search button.
- Step 2: System return search results or message if nothing found.
- Step 3: User can click returned results to view detail.

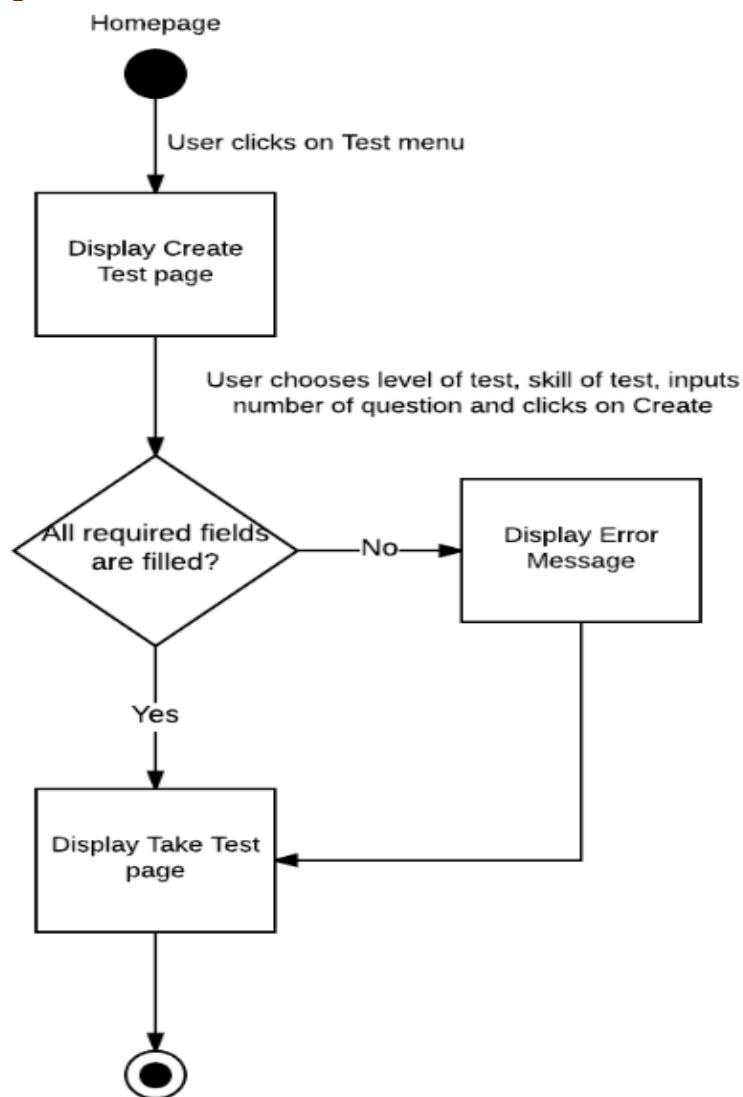
3.2.1.5 Study workflow



Brief description: Guest and Member can use this function. Steps:

- Step 1: User clicks on Courses menu.
- Step 2: System display List Lesson page.
- Step 3: User clicks on Study button.
- Step 4: System display Lesson Detail page.

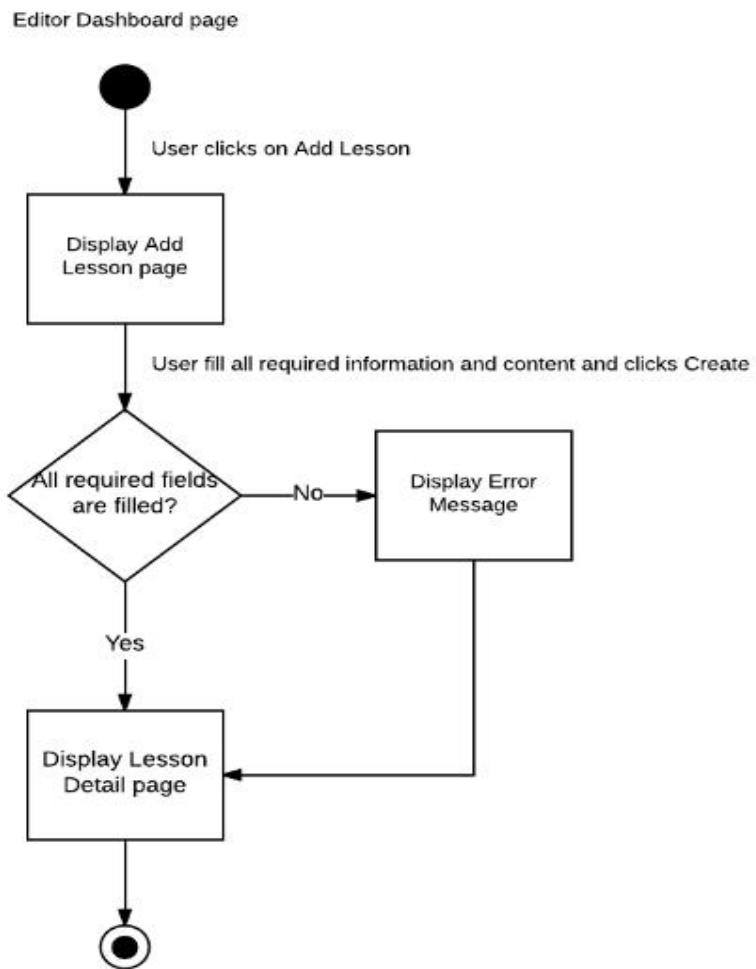
3.2.1.6 Testing workflow



Brief description: Guest and Member can use this function. Steps:

- Step 1: User clicks on Test menu.
- Step 2: System display Create Test page.
- Step 3: User chooses level of test, skill test, and number of questions.
- Step 4: User clicks on Create button.
- Step 5: System display Take Test page.

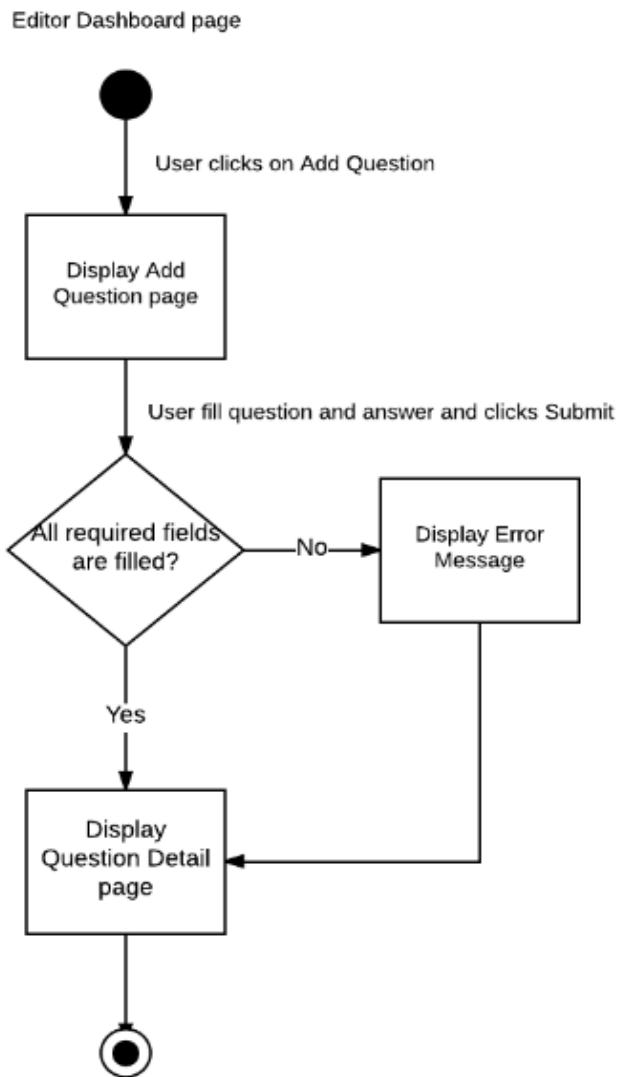
3.2.1.7 Add Lesson workflow



Brief description: Content Editor can use this function. Steps:

- Step 1: Content Editor clicks on Add Lesson in Lesson menu.
- Step 2: System display Add Lesson page...
- Step 3: Content Editor fills all required information and content of lesson and clicks on Create button.
- Step 4: System display Lesson Detail page or Error Message.

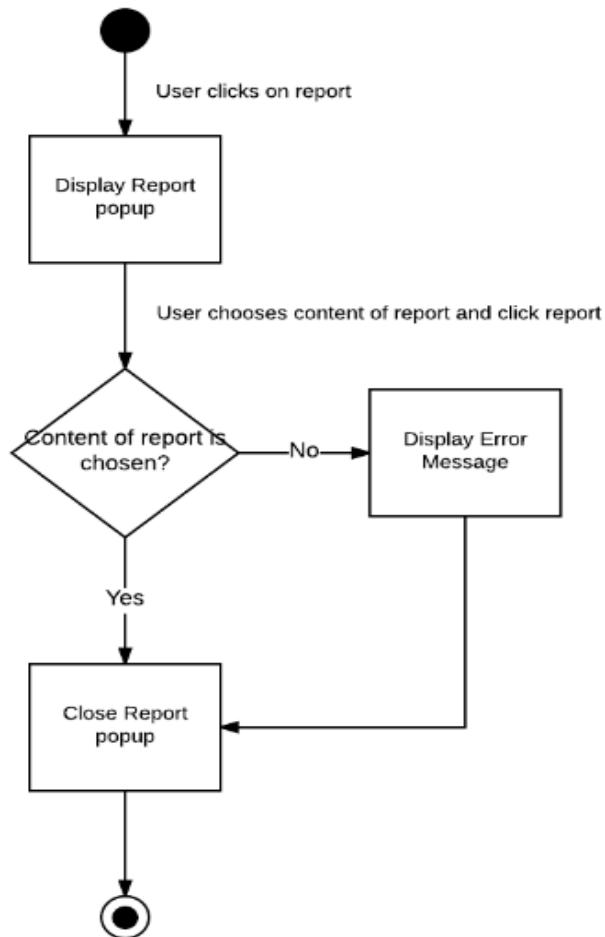
3.2.1.8 Add Question workflow



Brief description: Content Editor can use this function. Steps:

- Step 1: Content Editor clicks on Add Question in Question menu.
- Step 2: System display Add Question page...
- Step 3: Content Editor fills question and answer of question and clicks on Submit button.
- Step 4: System display Question Detail page or Error Message.

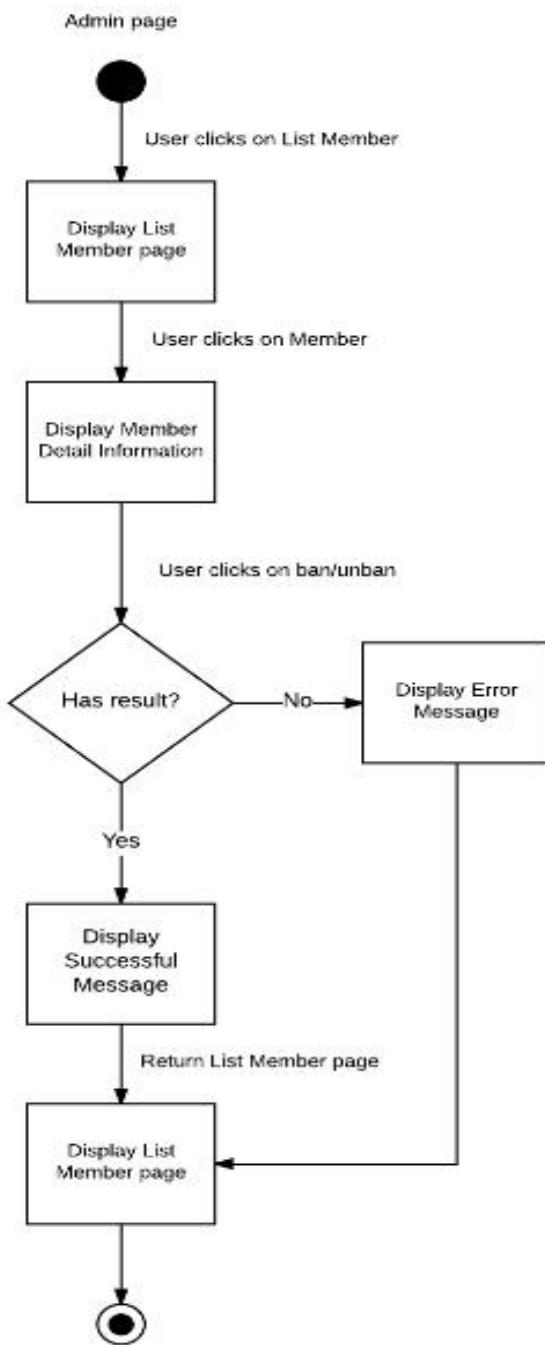
3.2.1.9 Report Content workflow



Brief description: Member can use this function. Steps:

- Step 1: Member clicks on Report button.
- Step 2: System display Report popup.
- Step 3: Member chooses content of report and clicks on Report.
- Step 4: System close Report popup or Error Message.

3.2.1.10 Ban/Unban user workflow



Brief description: Administrator can use this function. Steps:

- Step 1: Administrator clicks on List Member menu.
- Step 2: System display List Member page.
- Step 3: Administrator clicks on Member who Administrator want to ban/unban.
- Step 4: System display Member Detail Information page.

- Step 5: Administrator clicks on Ban/Unban button.
- Step 6: System display Successful Message or Error Message.
- Step 7: System display List Member page.

3.2.2 Product Features

Veazy is a website, user only connect to internet to use website. System provides these main functions:

3.2.2.1 Guest features

- **Register:** User can register new account and login to use all features of Veazy.
- **Login:** User login to Veazy system.
- **Search:** User can search lessons, words and system will display all of result related to keywords.
- **Forgot Password:** If user forgot their password, user can send a request to reset their password.
- **Learning:** User can learn Vietnamese through lessons of system.
- **Testing:** User can take a multiple-choice test and view result at the end of test.

3.2.2.2 Member features

- **Management Profile:** System just allows user who logins as Member can manage their profile such as view their information, update their information, change new password, as well as view their statistical learning process.
- **Report Content:** System just allows user as Member can send a report to Content Editor about wrong content of lesson or question.
- **Retake Test:** System just allows user as Member can view their list test history, review test and retake test.

3.2.2.3 Content Editor features

- **Management Profile:** When user logins as a Content Editor, user can manage their profile such as view Content Editor Information, update information and change password.
- **Manage Lesson:** When user logins as a Content Editor, user can manage lesson of system. So that user can view all lesson of system, search, add new lesson, edit or delete lesson.
- **Manage Question:** When user logins as a Content Editor, user can manage question of system. So that user can view all question of system, search, add new question, edit or delete question.
- **View Report:** System's Member send reports about wrong content of lesson or question. Content Editor can view this reports to determined that the content is wrong or not and edit it.

3.2.2.4 Admin features

- **Management Profile:** When user logins as an Admin, user can manage their profile such as view Admin information and change password.
- **Manage user:** When user logins as an Admin, user can manage member of system as: view list member of system, view detail information, ban/unban member or set permission for member.
- System statistics:** When user logins as an Admin, user can view statistics of system...

3.3 Software Requirements Specification

3.3.1 Overall Description

3.3.1.1 System overview

The product is a web-based system implementing client-server model. Veazy system serves as an e-learning system which targets mainly Japanese people who working in Vietnam and have demand for studying Vietnamese.

The purpose of Veazy is to create a friendly, easy-to-use and free-of-charge Vietnamese learning system with various features that improves efficiency in study significantly as well as provide a coherent system of ability assessment to help users keep track of their level in Vietnamese language...

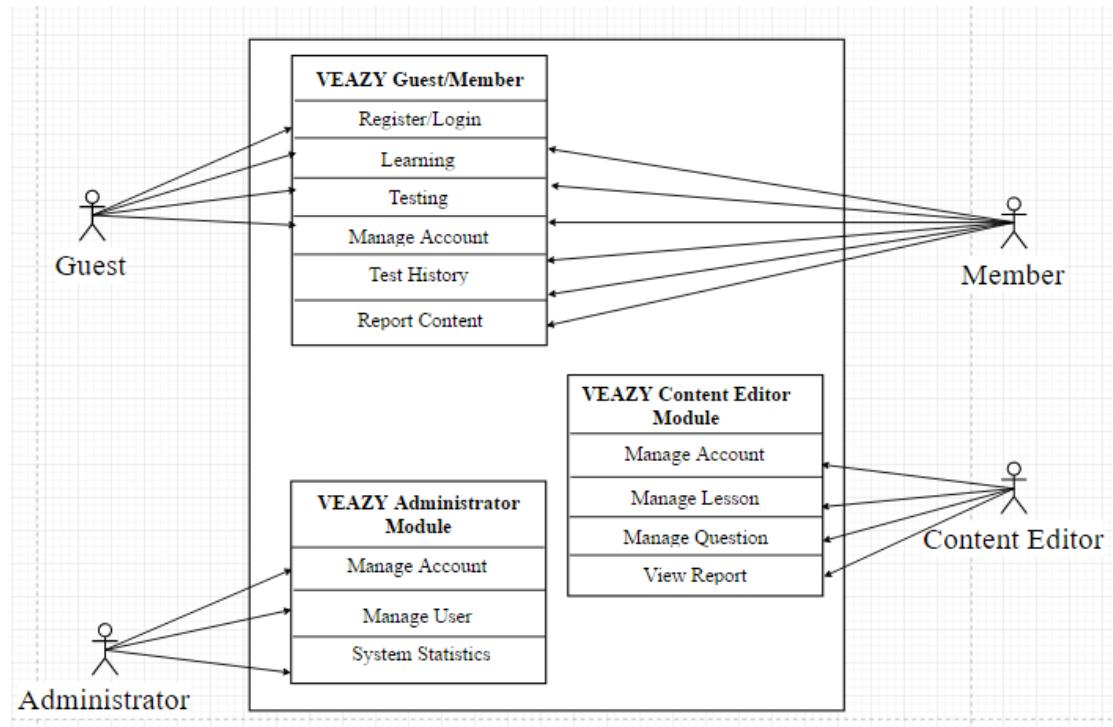


Figure 3-5: System overview

3.3.1.2 Product Features

The following are the main features that are included in the system:

- User Account Management: Allows users register new accounts in order to log into the system and provide features of updating and viewing profiles.
- E-learning System: Provides a well-organized lessons system divided in courses based on level of difficulty.
- Taking Test: Allows users to take small test and display test result.

Main functions of Users:

User	Function Name	Description
Guest	Register new Account	User registers new account to log into the system
	Login	User login to the system.
	View Lesson List	User views list lessons of courses.
	View Lesson Detail	User views the content of the selected lesson.
	Take Test	User takes a multiple-choice test.
	View Test Result	User views result of their test.
Member	View Profile	User views their own profile.
	Edit Profile	User edits their own profile.
	Change Password	User changes their password
	View Test Statistic	User views summary about their test and their score
	Logout	User logs out of system.
	Report Lesson	User sends their report to Content Editor about wrong content of lesson
	Report Question	User sends their report to Content Editor about wrong content of question
	View List Test History	User views list test history
	View Test History	User views test history detail
	Retake Test	User retake test
Content Editor	View Profile	User views their own profile
	Edit Profile	User edits their own profile
	Change Password	User changes their password
	Logout	User logs out of system.
	View Lesson List	User views list lessons of courses.

	View Lesson Detail	User views the content of the selected lesson.
	Filter Lesson	User filters lesson list
	Create Lesson	User creates new lesson
	Edit Lesson	User edits lesson
	Delete Lesson	User deletes lesson
	View Question List	User views list question
	View Question Detail	User views question detail
	Filter Question	User filters question list
	Create Question	User creates new question
	Edit Question	User edits question
	Delete Question	User deletes question
	View Report	User views report about wrong content
Admin	View Profile	User views their own profile
	Change Password	User changes their password
	Logout	User logs out of system.
	View User List	User views list member of system
	View User Profile	User views member's profile.
	Set Role User	User sets role for member's system.
	Ban User	User bans member out of system.
	Unban User	User unbans member.
	View System Statistic	User views statistic of system.
	Search User	User searching for user.

Table 3-1: Product features

3.3.1.3 User Characteristics

There are 3 types of users involving in the system:

- Normal users (Guest, Member): Requires basic knowledge of internet & computer.
- Content Editor: Required basic knowledge of internet & computer as normal user. In addition to that, they need to have upper-level of Vietnamese language and ability write academic articles as well as managing them.
- Administrators: Required to have high-level technical, management skills and the ability to promote the website.

3.3.1.4 User Documentation

List of document when delivering this application is in the table below:

Name	Description
User Manual	Provide detailed explanation about the system, screens and guide users how to use all features of VEAZY website.

3.3.1.5 Assumption

The system depends on which type and version of browser being installed on the client machine i.e. the browser should support both HTML5 and CSS3. The internet connection speed and bandwidth is also a dependency of the system.

One assumption that user has a strong enough internet connect which can adapt to system load. If the speed and bandwidth is slow, there is one scenario that browser cannot fully load the webpage, which may affect web layout displaying. Another scenario is that when user plays media such as listening or watching conversation, the media quality may be affected badly by a weak internet connection, which results a bad user experience.

Another assumption is that user is using a fully technical supported browser. If the browser does not support full technique, website cannot adapt full functionalities...

3.3.2 Entity Relation Model

3.3.2.1 Entity-Relationship Diagram

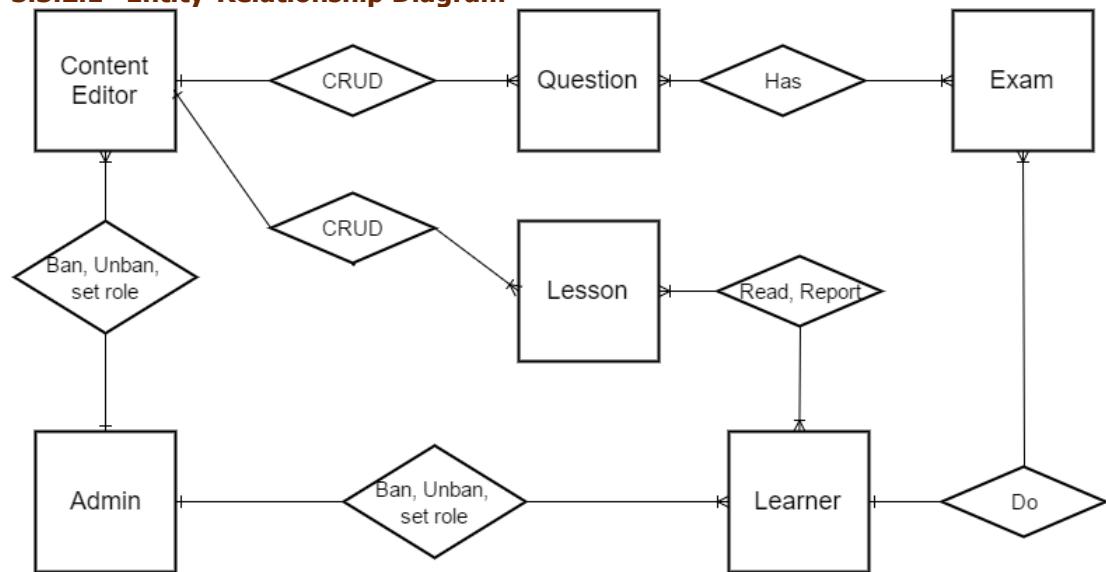
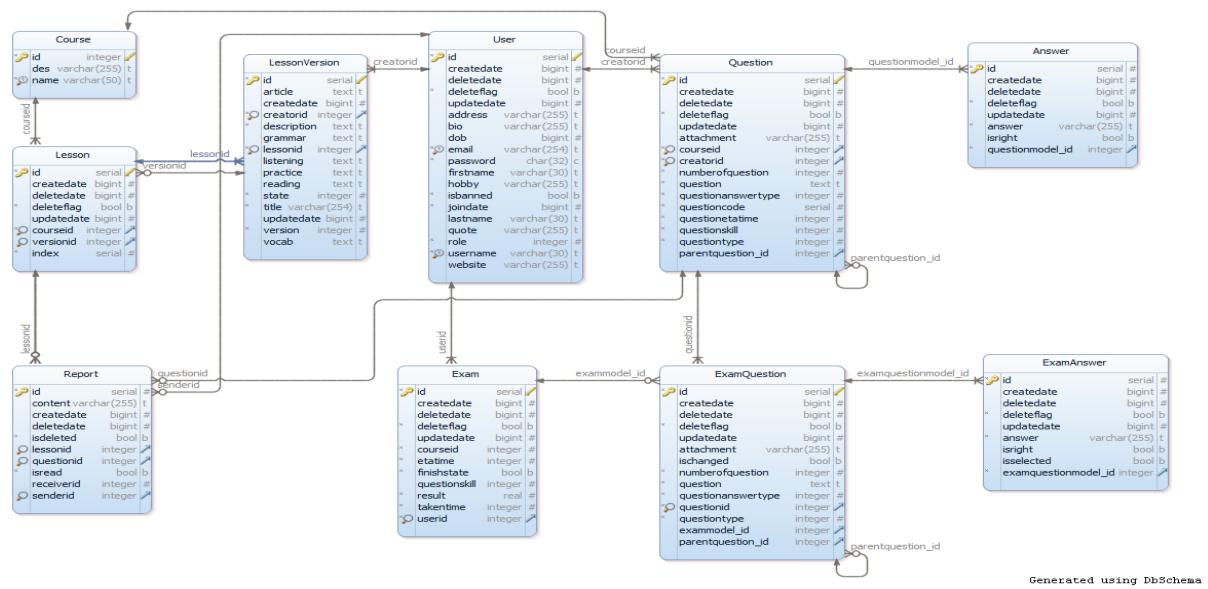


Figure 3-6: Entity Relationship Diagram

No	Entity	Description
1	Admin	System administrator
2	Learner	The end-user who learn on the website
3	Content Editor	The user who create, update lesson content and questions.
4	Question	Question for the exam
5	Lesson	The lesson
6	Exam	Examination for the user

3.3.2.2 Table Details

3.3.2.2.1 Table diagram



Generated using DbSchema

Figure 2: Table diagram

3.3.2.2.2 Course table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	integer		NOT		P		
2	des	character	255					description
3	name	character	50	NOT	U			

3.3.2.2.3 Exam table

No	Field Name	Type	Null	Unique	P/F Key	Default	Description
1	id	serial	NOT		P		
2	createdate	bigint					
3	deletedate	bigint					
4	deleteflag	bool	NOT			FALSE	
5	updatedate	bigint					
6	courseid	integer	NOT		F		
7	etetime	integer	NOT			0	estimate time
8	finishstate	bool	NOT				
9	questionskill	integer					
10	result	real	NOT			0.0	
11	takentime	integer	NOT			0	
12	userid	integer	NOT		F		

3.3.2.2.4 Lesson table

No	Field Name	Type	Null	Unique	P/F Key	Default	Description
1	id	serial	NOT		P		
2	createdate	bigint					
3	deletedate	bigint					
4	deleteflag	bool	NOT			FALSE	
5	updatedate	bigint					
6	courseid	integer	NOT		F		
7	versionid	integer			F		
8	index	serial	NOT				

3.3.2.5 LessonVersion table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		
2	article	text						
3	createdate	bigint						
4	creatorid	integer		NOT		F		
5	description	text		NOT				
6	grammar	text						
7	lessonid	integer		NOT		F		
8	listening	text						
9	practice	text						
10	reading	text						
11	state	integer		NOT			1	
12	title	varchar	254	NOT				
13	updatedate	bigint						
14	version	integer		NOT			1	
15	vocab	text						

3.3.2.6 Report table

No	Field Name	Type	Null	Unique	P/F Key	Default	Description
1	id	serial	NOT		P		
2	createdate	bigint					
3	deletedate	bigint					
4	deleteflag	bool	NOT			FALSE	
5	updatedate	bigint					
6	lessonid	integer			F		
7	questionid	integer			F		
8	isread	bool	NOT			FALSE	
9	receiverid	integer			F		
10	senderid	integer			F		

3.3.2.7 Answer table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		Auto increase
2	createdate	bigint						
3	deletedate	bigint						
4	deleteflag	bool		NOT			FALSE	
5	updatedate	bigint						
6	answer	varchar	255	NOT				
7	isright	bool		NOT			FALSE	
8	questionmodel_id	integer		NOT		F		

3.3.2.8 User table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		Auto increase
2	createdate	bigint						
3	deletedate	bigint						
4	deleteflag	bool		NOT			FALSE	
5	updatedate	bigint						
6	address	varchar	255					
7	bio	varchar	255					biographic
8	dob	bigint						date of birth
9	email	varchar	254	NOT	U			
10	password	char(32)		NOT				
11	firstname	varchar	30					
12	hobby	varchar	255					
13	isbanned	bool		NOT			FALSE	
14	joindate	bigint		NOT				
15	lastname	varchar	30					
16	quote	varchar	255					
17	role	integer		NOT			3	
18	username	varchar	30		U			
19	website	varchar	255					

3.3.2.2.9 ExamAnswer table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		Auto increase
2	createdate	bigint						
3	deletedate	bigint						
4	deleteflag	bool		NOT			FALSE	
5	updatedate	bigint						
6	answer	varchar	255	NOT				
7	isright	bool		NOT			FALSE	
8	isselected	bool		NOT			FALSE	
9	examquestion model_id	integer				F		

3.3.2.2.10 ExamQuestion table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		Auto increase
2	createdate	bigint						
3	deletedate	bigint						
4	deleteflag	bool		NOT			FALSE	
5	updatedate	bigint						
6	attachment	varchar	255					
7	ischanged	bool						
8	numberofquestion	integer		NOT			1	
9	question	text		NOT				
10	questionanswertype	integer		NOT			1	
11	questionid	integer		NOT		F		
12	questiontype	integer		NOT			1	
13	exammodel_id	integer				F		
14	parentquestion_id	integer				F		

3.3.2.2.11 Question table

No	Field Name	Type	Max Length	Null	Unique	P/F Key	Default	Description
1	id	serial		NOT		P		Auto increase
2	createdate	bigint						
3	deletedate	bigint						
4	deleteflag	bool		NOT			FALSE	
5	updatedate	bigint						
6	attachment	varchar	255					
7	courseid	integer		NOT		F		
8	creatorid	integer		NOT		F		
9	numberofquestion	integer		NOT			1	
10	question	text		NOT				
11	questionanswertype	integer		NOT			1	
12	questioncode	serial		NOT				
13	questionatetime	integer		NOT			60	
14	questionskill	integer		NOT			1	
15	questiontype	integer		NOT			1	
16	parentquestion_id	integer				F		

3.3.3 Functionality

3.3.3.1 Guest

3.3.3.1.1 Manage Account

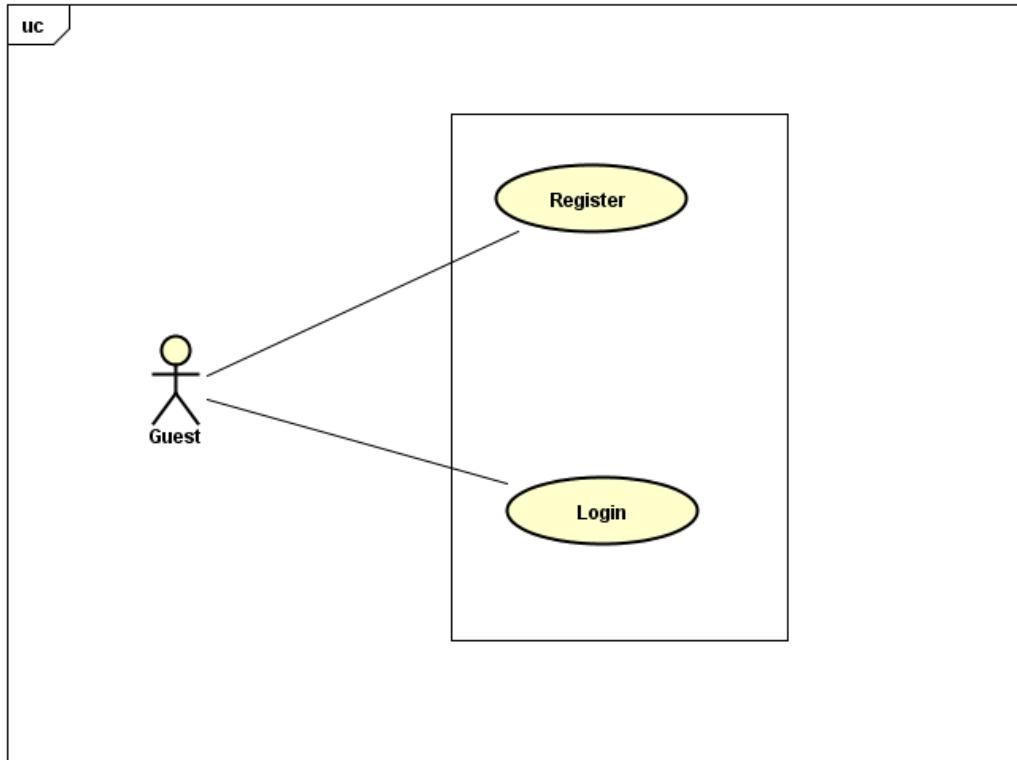


Figure 3-1 : UC: Manage account

3.3.3.1.1.1 UC-G-001: Login

Use Case Name	Log In		
Use Case ID	UC-G-001		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Guest		
Description	User to login to Veazy system		
Pre-condition	<ul style="list-style-type: none"> ● Website is available ● User has a registered account ● “Login” hyperlink is active at the right top of menu 		
Trigger	<ul style="list-style-type: none"> ● User inputs username and password into textboxes respectively in “Login” page ● User clicks “Login” button on “Login” page 		
Post Condition	<ul style="list-style-type: none"> ● Login successfully and redirect to “Home” page ● If exceptions are thrown, the result must match with output of each described exception. 		

Main Success Scenario

Step	User	System's Response
1	Click “Login” hyperlink at the right top of menu	Shows “Login” page
2	Enters username and password	Username and “Password” textbox border changes to green color
3	Clicks “Login” button	Redirects to “Home” page

Exception

At Step	User	System's Response
2	If “Username” textbox is null	Border of “Username” textbox changes into red Displays the message: “Username required!”

2	If "Password" textbox is null	Border of "Password" textbox changes into red Displays the message "Password required!"
3	If user enters the wrong password or username	Displays the message "Incorrect username or password!"

3.3.3.1.1.2 UC-G-002: Register

Use Case Name	Register		
Use Case ID	UC-G-002		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	29/07/2016
Actor	Guest		
Description	User registers new account to log into the system		
Pre-condition	<ul style="list-style-type: none"> Website is available 		
Trigger	<ul style="list-style-type: none"> User clicks "Sign up" button on "Home" page User inputs username, email, password and confirms password User clicks "Register" button on "Register" page 		
Post Condition	<ul style="list-style-type: none"> Register successfully and redirect to "Register Success" page If exceptions are thrown, the result must match with output of each described exception. 		

Main Success Scenario

Step	User	System's Response
1	Clicks "Sign up" button on "Home" page	Shows "Register" page
2	Inputs username, email, password and password confirmation	
3	Clicks "Register" button	Redirects to "Register Success" page

Alternative Scenario		
At Step	User	System's Response
Exception		
At Step	User	System's Response
2	If "Username" textbox is null	<p>Border of "Username" textbox changes to red color.</p> <p>Displays the message: "Username required!"</p>
2	If "Email" textbox is null	<p>Border of "Email" textbox changes to red color.</p> <p>Displays the message: "Email required!"</p>
2	If "Password" textbox is null	<p>Border of "Password" textbox changes to red color</p> <p>Displays the message: "Password required!"</p>
2	If user inputs username that is less than 6 characters	<p>Border of "Username" textbox changes to red color</p> <p>Displays the message: "Username must be at least 6 characters long!"</p>
2	If user inputs username that is more than 30 characters	<p>Border of "Username" textbox changes to red color</p> <p>Displays the message: "Username must be no more than 30 characters long!"</p>
2	If user inputs invalid email	<p>Border of "Email" textbox changes to red color</p> <p>Displays the message: "Invalid email!"</p>
2	If user inputs password that is less than 6 characters	<p>Border of "Password" textbox changes to red color</p>

		Displays the message: "Password must be at least 6 characters long!"
2	If user inputs password that is more than 32 characters	Border of "Password" textbox changes to red color Displays the message: "Password must be no more than 32 characters long!"
2	If the confirmed password is not the same with password	Border of "Confirm Password" textbox changes to red color Displays the message: "Password does not matched!"

3.3.3.1.2 Learning

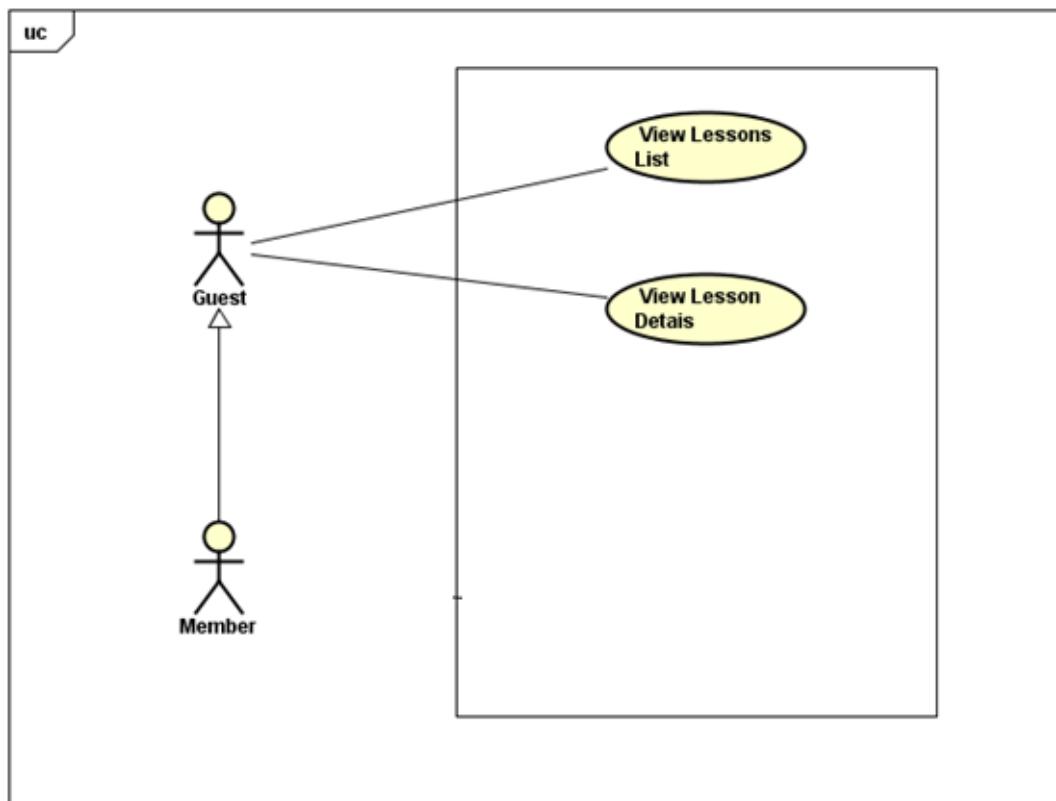


Figure 3-2 : UC: Learning

3.3.3.1.2.1 UC-G-003: View Lessons List

Use Case Name	View Lessons List		
Use Case ID	UC-G-003		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016
Actor	Guest, Member		
Description	User views list lessons of courses		
Pre-condition	<ul style="list-style-type: none"> • Website is available • “Courses” hyperlink is active in top menu • Course level hyperlink in dropdown list is active 		
Trigger	User clicks “Course” link on top menu then clicks the hyperlink of the course level that user wants to choose in dropdown list		
Post Condition	System redirects to “Course” page which displays lesson list in selected course level		
Main Success Scenario			
Step	User	System's Response	
1	Clicks “Course” link on top menu	Shows level dropdown list	
2	Clicks a course level hyperlink in dropdown list	Redirect to “Courses” page and displays lesson list in the selected course	

3.3.3.1.2.2 UC-G-004: View Lesson Details

Use Case Name	View Lesson Details		
Use Case ID	UC-G-004		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Guest, Member		
Description	User views the content of the selected lesson		
Pre-condition	Website is available		
Trigger	User clicks "Study" button that belongs to the lesson user wants to study on "Course" page		
Post Condition	System redirects to "Lesson" pagewhich displays the content of the selected lesson		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Study" button of a lesson on "Course" page	Redirects to "Lesson" page and displays lesson content	

3.3.3.1.3 Testing

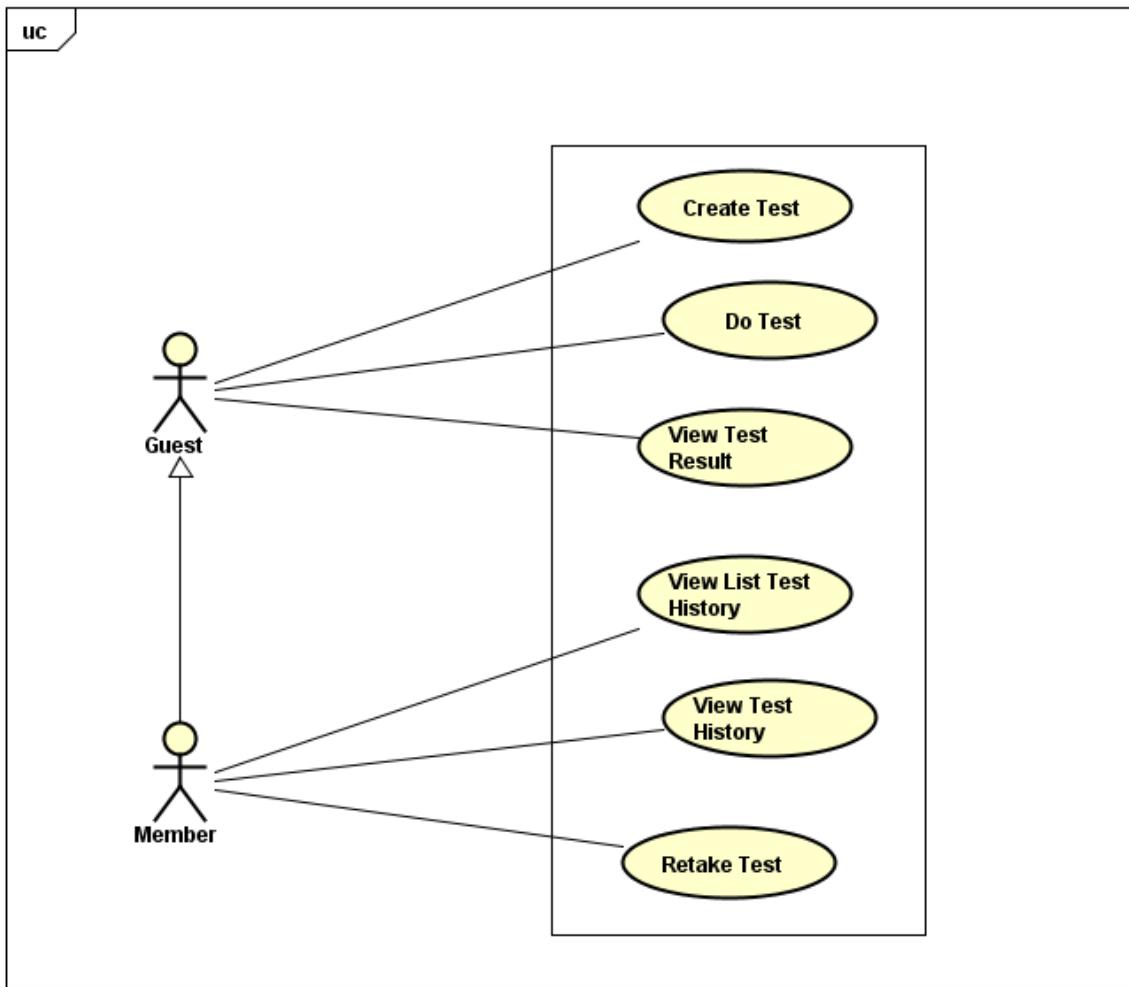


Figure 3-3 UC: Testing

3.3.3.1.3.1 UC-G-005: Create Test

Use Case Name	Create Test		
Use Case ID	UC-G-005		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	14/08/2016
Actor	Guest, Member		
Description	User setups to create a new test		
Pre-condition	Website is available		
Trigger	<ul style="list-style-type: none"> • User clicks "Test" link at top menu • User clicks "Level" dropdown list to select difficulty level • User clicks "Test skill" dropdown list to select the skill that user wants to test • User clicks the upward/downward arrow of "Number of question" spin box to increase/decrease the number of questions in test • User clicks "Create" button on "Test" page 		
Post Condition	<ul style="list-style-type: none"> • System redirects to "Test" page and displays test content which contains the questions in accordance with user's setup • If exceptions are thrown, the result must match with output of each described exception. 		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Test" link at top menu	Redirects to "Test" page	
2	Clicks "Level" dropdown list to select difficulty level		
3	Clicks "Test skill" dropdown list to select test skill		
4	Clicks the upward/downward arrow of "Number of question"	The number of questions goes up/down	
5	Clicks "Create" button on "Test" page	System shows test content	

Exception		
At step	User	System's response
5	If there is no questions that can be generated in accordance with user's selected level, skill	Displays the message: "No question of this level & skill are added yet!"

3.3.3.1.3.2 UC-G-006: Do Test

Use Case Name	Do Test		
Use Case ID	UC-G-006		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	19/07/2016
Actor	Guest, Member		
Description	User do test		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-G-005 successfully 		
Trigger	<ul style="list-style-type: none"> • User clicks the checkbox of the answer that user think is the right answer of a question. 		

Main Success Scenario

Step	User	System's Response
1	User clicks the checkbox of answers of questions	The checkbox value changes to "true/false" value, indicating the answer(s) selected by user

3.3.3.1.3.3 UC-G-007: View Test Result

Use Case Name	View Exam Result		
Use Case ID	UC-G-007		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	19/07/2016

Actor	Guest, Member	
Description	User submit test and views result	
Pre-condition	<ul style="list-style-type: none"> Website is available 	
Trigger	<ul style="list-style-type: none"> User clicks “Check Answer” button at bottom test content on “Taking Test” page to submit the test. User can also wait for time duration to run out so that the test is automatically submitted. 	
Post Condition	System redirects to “Test Result” page which displays test result	
Main Success Scenario		
Step	User	System's Response
1	Clicks “Check Answer” button	Redirects to “Test Result” page and displays test result

3.3.3.1.4 Change Interface Language

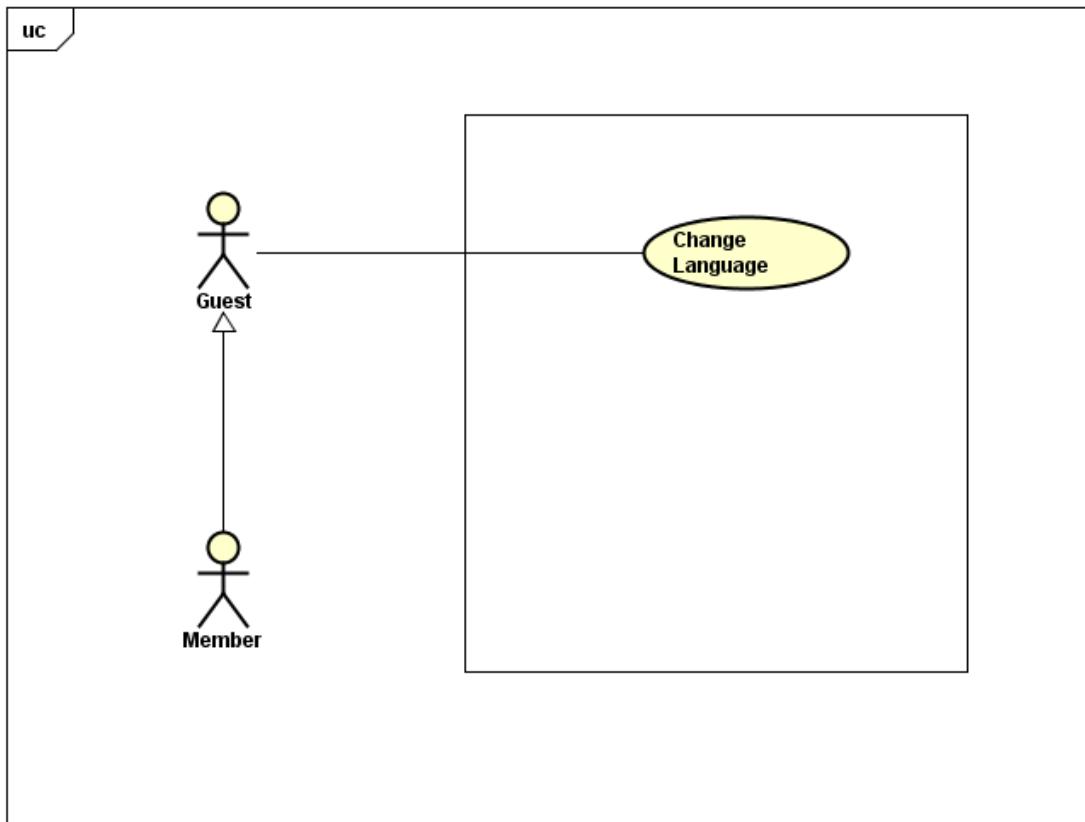


Figure 3-4: UC: Change Interface Language

UC-G-008: Change Language

Use Case Name	Change Language		
Use Case ID	UC-G-008		
Created By	ThangPD	Last Updated By	ThangPD
Date Created	27/07/2016	Date Last Updated	05/08/2016
Actor	Member		
Description	User changes language of website		
Pre-condition	Website is available		
Trigger	User clicks "Veazy" logo at the top of menu of the webpage		
Post Condition	System changes language of website to Japanese (if current language is English) or English (if current language is Japanese)		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Veazy" logo	System changes language of website to Japanese/English	

3.3.3.2 Member

3.3.3.2.1 Manage Account

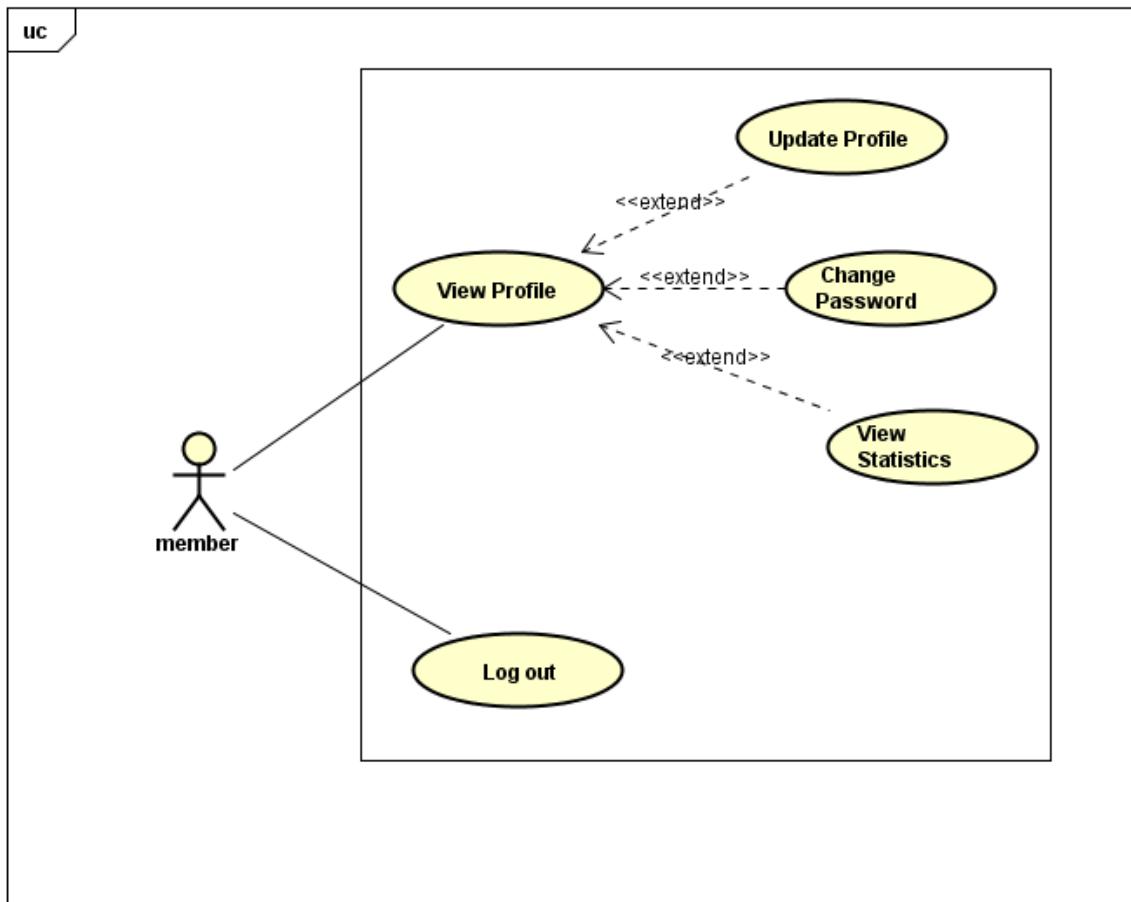


Figure 3-5. UC Manage Account

3.3.3.2.1.1 UC-M-001: View Profile

Use Case Name	View Profile		
Use Case ID	UC-M-001		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User views profile information		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system 		
Trigger	User clicks username hyperlink at the right corner of top menu of website		
Post Condition	System redirects to "User Profile" page		
Main Success Scenario			
Step	User	System's Response	
1	User clicks username hyperlink at the right corner of top menu	Redirects to "User Profile" page	

3.3.3.2.1.2 UC-M-002: Update Profile

Use Case Name	Update Profile		
Use Case ID	UC-M-002		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User edits information of their own profile		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-M-001 successfully 		

Trigger	<ul style="list-style-type: none"> User clicks "Update Profile" in "Actions" dropdown list User inputs information (including first name, last name, birthday, address, hobbies, about me, favorite quote, website) User clicks "Save" button at the bottom of "Update Profile Information" popup 	
Post Condition	<ul style="list-style-type: none"> System closes dialog, reloads "User Profile" page and show update successfully If exceptions are thrown, the result must match with output of each described exception. 	
Main Success Scenario		
Step	User	System's Response
1	Clicks "Update Profile" in "Actions" dropdown list	Shows "Update Profile Information" popup
2	Inputs first name, last name, birthday, address, hobbies, about me, favorite quote, website in respective textbox	
3	Click "Save" button.	Updated information of user, then closes popup and reloads "User Profile" page
Alternative Scenario		
At step	User	System's Response
3	Clicks "Cancel" button	Cancels editing profile action and closes "Update Profile Information" popup
Exception		
At Step	User	System's Response
2	If any of those textbox is null	System skips and does not update the information of the textbox

3.3.3.2.1.3 UC-M-003: Change Password

Use Case Name	Change Password		
Use Case ID	UC-M-003		
Created By	QuanNH	Last Updated By	ThangPD

Date Created	27/06/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User changes their password		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-M-001 successfully 		
Trigger	<ul style="list-style-type: none"> • User click "Change Password" in "Actions" dropdown list • User inputs current password, new password and re-type new password • User clicks "Change" button at the bottom of "Change Password" popup 		
Post Condition	<ul style="list-style-type: none"> • System stays on initial screen and password is changed successfully. • If exceptions are thrown, the result must match with output of each described exception. 		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Change Password" in "Actions" dropdown list.	Shows "Change password" popup	
2	Inputs current password, new password and password confirmation		
3	Clicks "Change" button.	Changes password successfully and closes popup	
Alternative Scenario			
At Step	User	System's Response	
3	Clicks "Cancel" button	Cancels changing password action and closes "Change Password" popup	
Exception			
At Step	User	System's Response	
2	If "Current Password" textbox is null	Displays the message: "Password required!"	
2	If "New Password" textbox is null	Displays the message: "Password required!"	

2	If new password and re-typed password is not the same	Displays the message: "Password does not match"
2	If user inputs new password that is less than 6 characters	Displays the message: "Password must be at least 6 characters long!"
2	If user inputs new password that is more than 32 characters	Displays the message: "Password must be no more than 32 characters long!"
2	If current password is incorrect	Displays the message: "Incorrect password!"

3.3.3.2.1.4 UC-M-005: View Statistics

Use Case Name	View Statistics		
Use Case ID	UC-M-005		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User views statistics assessed based on their test marks		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-M-001 successfully 		
Trigger	User clicks "Statistics" hyperlink on "User Profile" page		
Post Condition	System shows statistics successfully		

Main Success Scenario

Step	User	System's Response
1	Click "Statistics" button in "User Profile" page	System displays 3 types of statistics: radar chart of level, line chart of 10 recently taken tests and bar chart of skills.

3.3.3.2.1.5 UC-M-006: Logout

Use Case Name	Logout		
Use Case ID	UC-M-006		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Member		
Description	User logs out of Veazy system		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system 		
Trigger	User clicks username hyperlink at the right top of menu then clicks "Logout" hyperlink in dropdown list		
Post Condition	Redirect to "Login" page and signs user out successfully		
Main Success Scenario			
Step	User	System's Response	
1	Clicks username hyperlink at the right top of menu	Shows dropdown list	
2	Clicks "Logout" hyperlink in dropdown list	Signs user out, then redirects to "Login" page	

3.3.3.2.2 Report Content

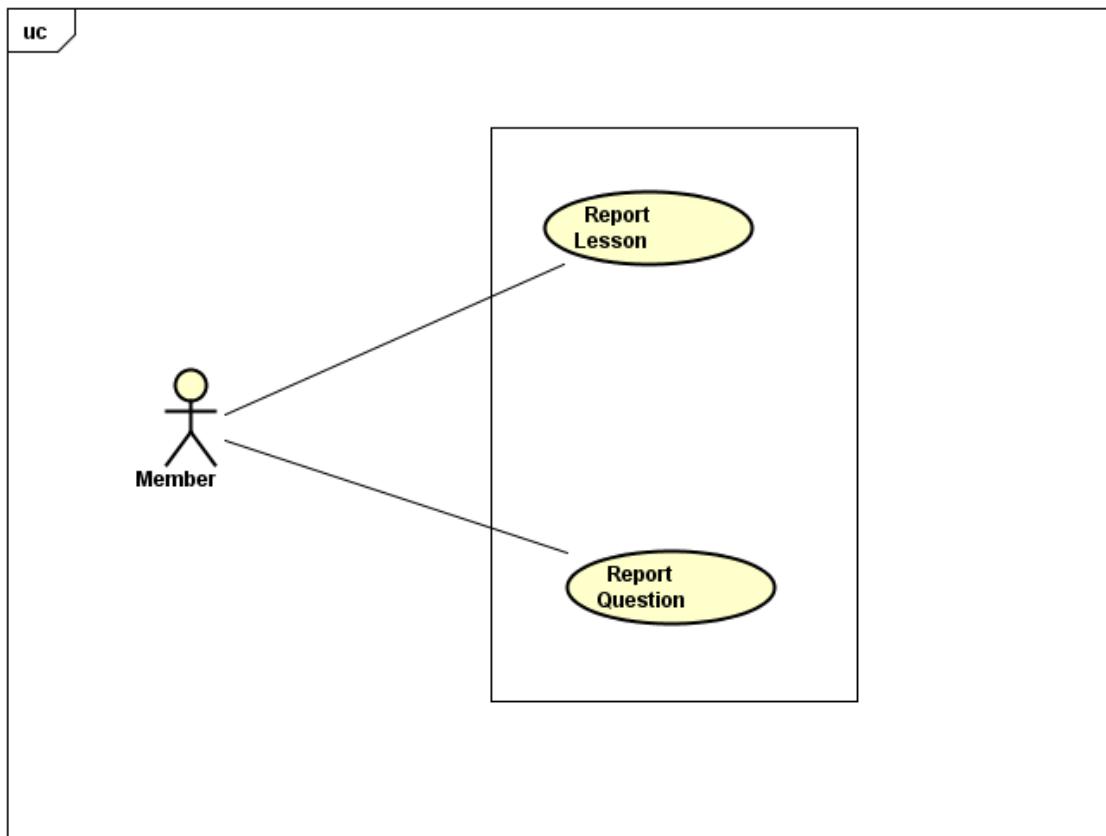


Figure 3-6. UC Report Content

3.3.3.2.2.1 UC-M-007: Report Lesson

Use Case Name	Report Lesson		
Use Case ID	UC-M-007		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	14/08/2016
Actor	Member		
Description	User sends reports about wrong content of lesson to Content Editor		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-G-004 successfully 		

Trigger	User clicks "Report" hyperlink on "Lesson" page	
Post Condition	System sends the report to Content Editor for review purpose	
Main Success Scenario		
Step	User	System's Response
1	Clicks "Report" hyperlink on "Lesson" page	Shows "Report Lesson" popup
2	Inputs content of report (i.e, the reason why user reports the lesson)	
2	Clicks "Report" button on "Report Lesson" popup	Sends report to Content Editor for review, then displays the message "Your report has been submitted for review. Thank you for your contribution."
Alternative Scenario		
At Step	User	System's Response
1	Clicks "Cancel" button on "Report lesson" popup	Cancels reporting lesson and closes the popup

3.3.3.2.2.2 UC-M-008: Report Question

Use Case Name	Report Question		
Use Case ID	UC-M-008		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User sends reports about wrong content of question to Content Editor		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-G-006 successfully 		
Trigger	User clicks "Report" button at the right side of the question that user wants to report on "Test Result" page		

Post Condition	System sends the report to Content Editor for review purpose	
Main Success Scenario		
Step	User	System's Response
1	Clicks "Report" button at the right of question	Shows "Report question" popup
2	Inputs content of report (i.e, the reason why user reports the question)	
3	Clicks "Report" button in "Report question" popup	Sends the report to Content Editor for review, then displays the message "Your report has been submitted for review. Thank you for your contribution."
Alternative Scenario		
At Step	User	System's Response
3	Clicks "Cancel" button on "Report question" popup	Cancels reporting question and closes the popup

3.3.3.2.3 Test History

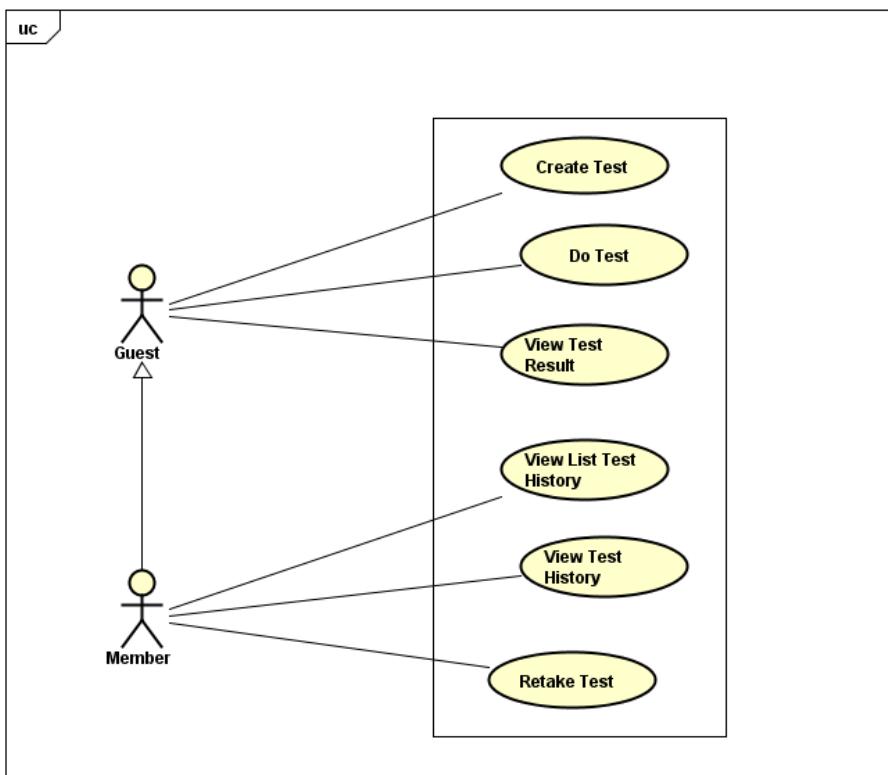


Figure 3-7. UC Test History

3.3.3.2.3.1 UC-M-009: View List Test History

Use Case Name	View List Test History		
Use Case ID	UC-M-009		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	20/07/2016
Actor	Member		
Description	User views list of their taken test (aka test history)		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system 		
Trigger	User clicks "Test History" tab on "User Profile" page		
Post Condition	Shows list test history in "Test History" tab		

Main Success Scenario		
Step	User	System's Response
1	Clicks "Test History" tab on "User profile" page	Shows a list of taken test records and their details including skill of test, course level, time allowed & time taken, result.

3.3.3.2.3.2 UC-M-010: View Test History

Use Case Name	View Test History		
Use Case ID	UC-M-010		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	29/07/2016
Actor	Member		
Description	User views the details of taken tests including all questions, answers (both true answers and their selected answers) as well as the test result		
Pre-condition	<ul style="list-style-type: none"> Website is available Member logged in the system Done UC-M-009 successfully 		
Trigger	User clicks "See result" in "Actions" dropdown list in user profile page		
Post Condition	System redirects to "Test Result" page		

Main Success Scenario

Step	User	System's Response
1	Clicks "See result" in "Actions" dropdown list in user profile page	Shows "Test Details" page

3.3.3.2.3.3 UC-M-011: Retake Test

Use Case Name	Retake Test
Use Case ID	UC-M-011

Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	28/07/2016
Actor	Member		
Description	User takes the tests (in their test history list) one more time		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Member logged in the system • Done UC-M-009 successfully 		
Trigger	User clicks "Retake this test" in "Actions" dropdown list on "User Profile" page		
Post Condition	System redirects to "Taking Test" page		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Actions" hyperlink on the top right corner of the test record	Displays dropdown list	
2	Clicks "Retake this test" hyperlink	Replicates the tests and redirects to "Taking Test" page	
Alternative Scenario			
At Step	User	System's Response	
2	Click "Cancel" button		
Exception			
At Step	User	System's Response	

3.3.3.3 Content Editor

3.3.3.3.1 Manage Account

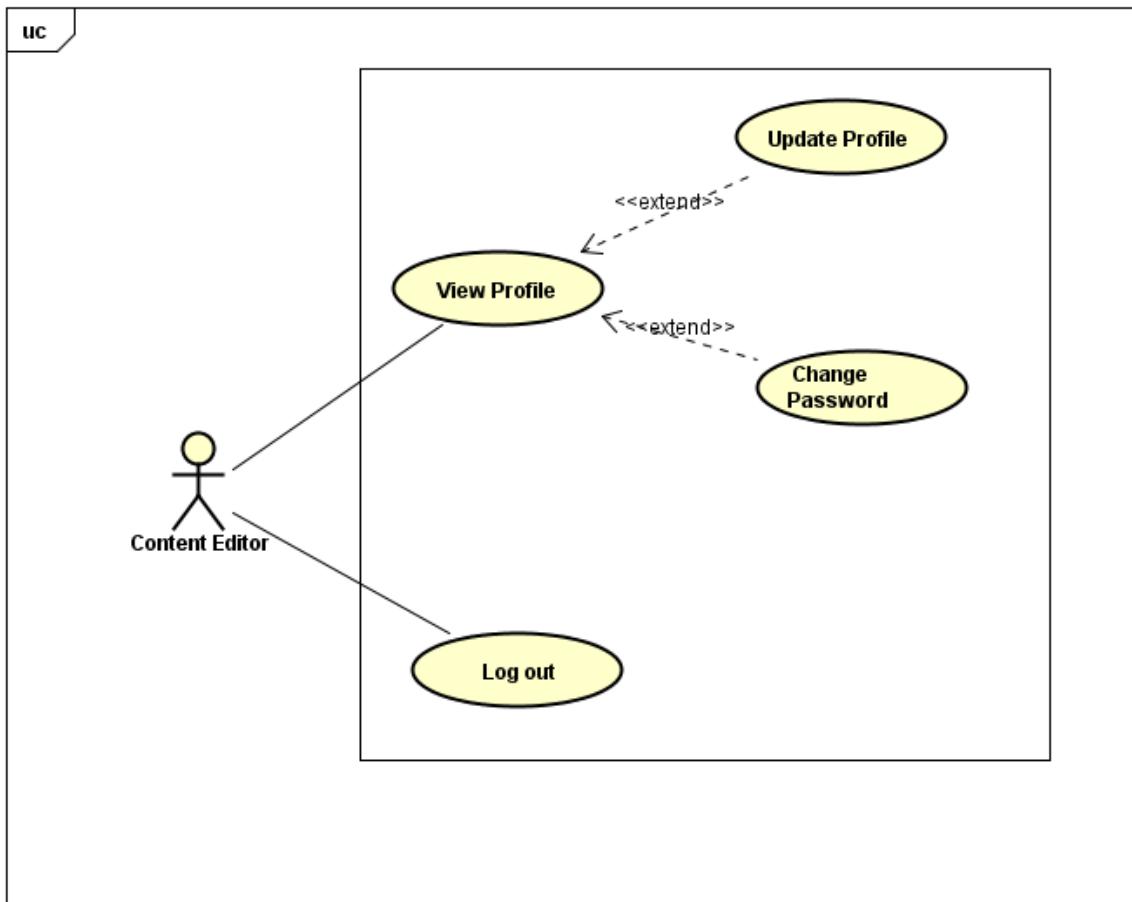


Figure 3-8. UC Manage Account

3.3.3.3.1.1 UC-E-001: View Profile

Use Case Name	View Profile		
Use Case ID	UC-E-001		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	29/07/2016
Actor	Content Editor		
Description	User views their profile		

Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system 	
Trigger	User clicks username hyperlink at right-sided editor menu	
Post Condition	System redirects to "Editor Profile" page	
Main Success Scenario		
Step	User	System's Response
1	Clicks username hyperlink at the right of avatar	Redirects to "Editor Profile" page

3.3.3.3.1.2 UC-E-002: Update Profile

Use Case Name	Update Profile		
Use Case ID	UC-E-002		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	29/07/2016
Actor	Content Editor		
Description	User edits information of their own profile		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system 		
Trigger	<ul style="list-style-type: none"> User clicks "Update Profile" in "Actions" dropdown list User inputs information (including first name, last name, birthday, address, hobbies, about me, favorite quote, website) User clicks "Save" button at the bottom of "Update Profile Information" popup 		
Post Condition	<ul style="list-style-type: none"> System closes the dialog, reloads "Editor Profile" page and shows update successfully If exceptions are thrown, the result must match with output of each described exception. 		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Update Profile" in "Actions" dropdown list	Shows "Update Profile Information" popup	
2	Inputs first name, last name, birthday, address, hobbies,		

	about me, favorite quote, website in respective textbox	
3	Clicks “Save” button	Updates profile successfully, closes dialog and reload “Editor Profile” page
Alternative Scenario		
At Step	User	System's Response
3	Clicks “Cancel” button.	Cancels editing profile action and closes “Update Profile Information” popup
Exception		
At Step	User	System's Response
2	If any of those textbox is null	System skips and does not update the information of the textbox

3.3.3.3.1.3 UC-E-003: Change Password

Use Case Name	Change Password		
Use Case ID	UC-E-003		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Content Editor		
Description	User changes their password		
Pre-condition	<ul style="list-style-type: none"> Website is available Content Editor logged in the system Done UC-E-001 successfully 		
Trigger	User clicks “Change password” hyperlink in “Actions” dropdown list on “Editor Profile” page		
Post Condition	<ul style="list-style-type: none"> User click “Change Password” in “Actions” dropdown list User inputs current password, new password and re-type new password User clicks “Change” button at the bottom of “Change Password” popup 		
Main Success Scenario			

Step	User	System's Response
1	Clicks "Change Password" in "Actions" dropdown list.	Show "Change Password" popup
2	Inputs current password, new password and password confirmation	
3	Click "Save" button.	Changes password successfully and closes popup
Alternative Scenario		
At Step	User	System's Response
1	Clicks "Cancel" button.	Cancels changing password action and closes "Change Password" popup
Exception		
At Step	User	System's Response
2	If "Current Password" textbox is null	Displays the message: "Password required!"
2	If "New Password" textbox is null	Displays the message: "Password required!"
2	If new password and re-typed password is not the same	Displays the message: "Password does not match"
2	If user inputs new password that is less than 6 characters	Displays the message: "Password must be at least 6 characters long!"
2	If user inputs new password that is more than 32 characters	Displays the message: "Password must be no more than 32 characters long!"
2	If current password is incorrect	Displays the message: "Incorrect password!"

3.3.3.3.1.4 UC-E-004: Logout

Use Case Name	Logout
Use Case ID	UC-E-004

Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Content Editor		
Description	User logouts of the Veazy system		
Pre-condition	<ul style="list-style-type: none"> ● Website is available ● Content editor logged in the system 		
Trigger	User clicks "Logout" hyperlink at right-sided editor menu		
Post Condition	Signs user out successfully and redirects to "Login" page		
Main Success Scenario			
Step	User	System's Response	
1	Click "Logout" hyperlink	System redirects to "Login" page	

3.3.3.3.2 Change Interface Language

3.3.3.3.2.1 UC-E-005: Change Language

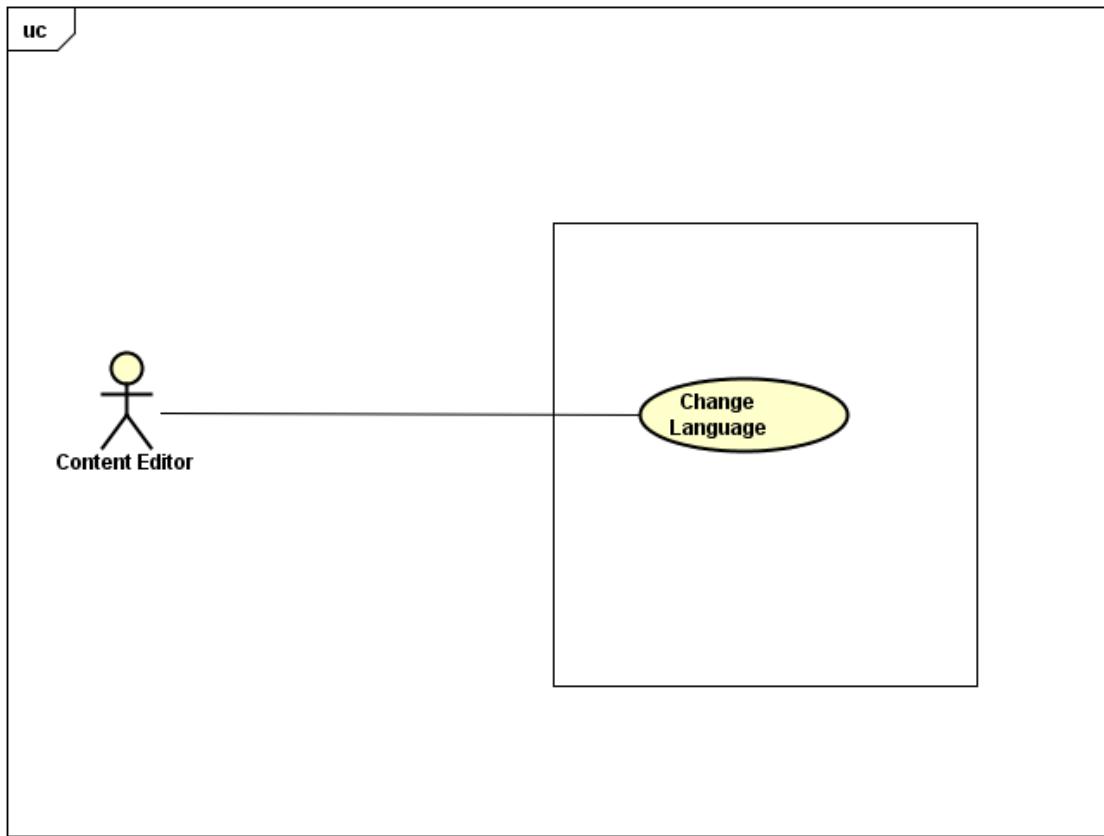


Figure 3-9. UC Change language

Use Case Name	Change Language		
Use Case ID	UC-E-005		
Created By	ThangPD	Last Updated By	ThangPD
Date Created	27/07/2016	Date Last Updated	05/08/2016
Actor	Content Editor		
Description	User changes language of website		
Pre-condition	Website is available		
Trigger	User clicks "Change Language To Japanese/English" hyperlink at right-sided editor menu		

Post Condition	System changes language of website to English/Japanese	
Main Success Scenario		
Step	User	System's Response
1	Click "Change Language to English/Japanese" hyperlink	System changes language of website to English/Japanese

3.3.3.3.3 Manage Lessons

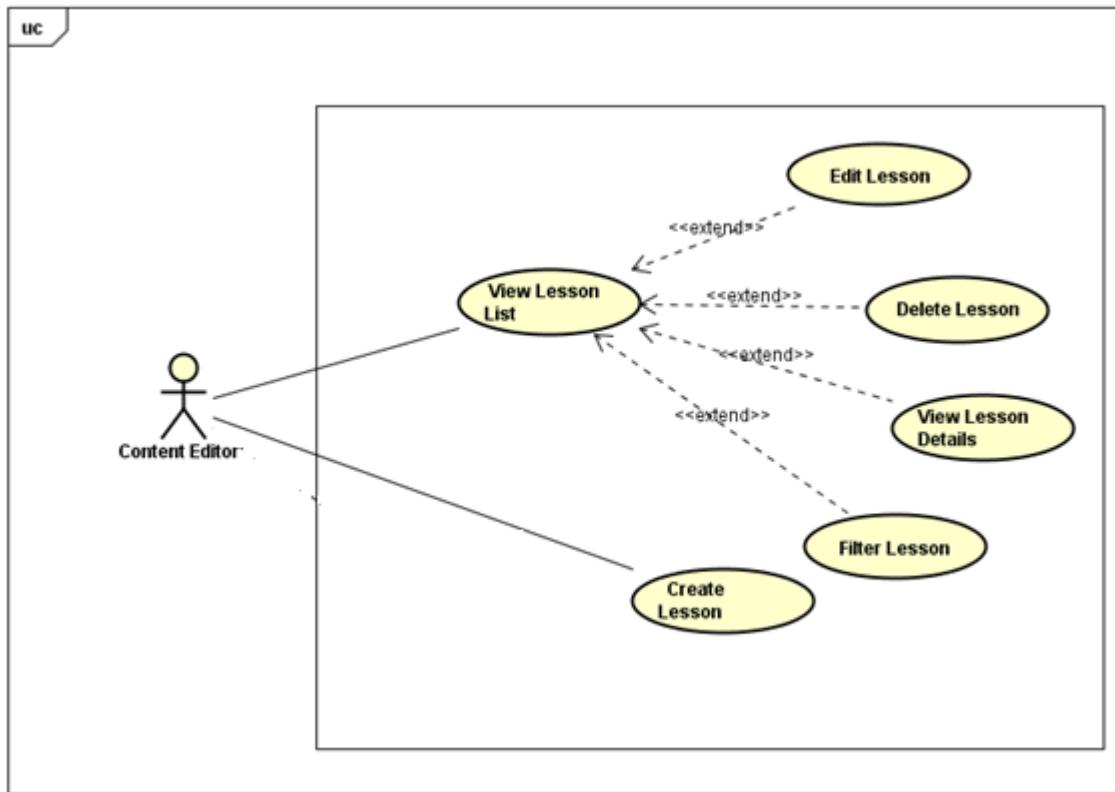


Figure 3-10. UC Manage Lessons

3.3.3.3.3.1 UC-E-006: View Lesson List

Use Case Name	View Lesson List		
Use Case ID	UC-E-006		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	05/08/2016
Actor	Content Editor		

Description	User views list lessons of courses		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system 		
Trigger	User clicks “Lessons” hyperlink at right-sided editor menu, then clicks “Lesson List” hyperlink		
Post Condition	System redirects to “Lesson List” page		
Main Success Scenario			
Step	User	System's Response	
1	Clicks “Lessons” hyperlink	Shows “Lesson List”hyperlink and “Add Lesson ” hyperlink	
2	Clicks “Lesson List” hyperlink	Redirect to “Lesson List” page	

3.3.3.3.3.2 UC-E-007:Filter lesson

Use Case Name	Search Lesson		
Use Case ID	UC-E-007		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	03/08/2016
Actor	Content Editor		
Description	User searches lessons in lesson list page		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system 		
Trigger	<ul style="list-style-type: none"> User chooses lesson level in “Level” dropdown list User type a keyword in “Keyword” textbox User clicks “Filter” button in lesson list page 		
Post Condition	Searches lessons successfully. System filters all lessons of the selected level and contains the keyword, then displays results		
Main Success Scenario			
Step	User	System's Response	
1	Choose lesson level in “Level” dropdown list		

2	Inputs keyword in "Keyword" textbox	
3	Clicks "Filter" button	System filters and displays result

3.3.3.3.3.3 UC-E-008: View Lesson Details

Use Case Name	View Lesson Details		
Use Case ID	UC-E-008		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	05/08/2016
Actor	Content Editor		
Description	User views lesson details		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system 		
Trigger	User clicks title name of the lessons in table that users want to view its detail		
Post Condition	System redirects to "Lesson Detail" page		
Main Success Scenario			
Step	User	System's Response	
1	Click title name link	Redirect to "Lesson Detail" page	

3.3.3.3.4 UC-E-009: Create Lesson

Use Case Name	Create Lesson		
Use Case ID	UC-E-009		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016
Actor	Content Editor		

Description	User creates new lesson	
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system 	
Trigger	User clicks “Lessons” hyperlink at right-sided editor menu, then clicks “Add Lesson” hyperlink	
Post Condition	<ul style="list-style-type: none"> • System redirects to “Add Lesson” page • If exceptions are thrown, the result must match with output of each described exception. 	
Main Success Scenario		
Step	User	System's Response
1	Clicks “Lessons” link	Shows “Lesson List” hyperlink and “Add Lesson” hyperlink
2	Clicks “Add Lesson” hyperlink	Redirects to “Add Lesson” page
3	Inputs content of lesson for each section including vocabulary, grammar, conversation, listening, practice and reading	
4	Clicks “Create” button	Creates new lesson successfully and redirects to “Lesson Detail” page
Exception		
At Step	User	System's Response
3	If “Title” textbox is null	Displays the message: “Failed to create new lesson! Please make sure you do not leave any section content & lesson information blank.”
3	If “Description” textbox is null	Displays the message: “Failed to create new lesson! Please make sure you do not leave any section content & lesson information blank.”
3	If any section of lesson is null or left blank	Displays the message: “Failed to create new lesson! Please make sure

		you do not leave any section content & lesson information blank."
--	--	---

3.3.3.3.3.5 UC-E-010: Edit Lesson

Use Case Name	Edit Lesson		
Use Case ID	UC-E-010		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016
Actor	Content Editor		
Description	User edits content of a lesson		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system 		
Trigger	User clicks "Edit" button of the lesson in table that user wants to edit		
Post Condition	<ul style="list-style-type: none"> • System edits successfully and redirects to "Lesson Detail" page • If exceptions are thrown, the result must match with output of each described exception. 		

Main Success Scenario

Step	User	System's Response
1	Clicks "Edit" button	Redirects to "Edit lesson" page
2	Inputs content of lesson for each section including vocabulary, grammar, conversation, listening, practice and reading	
3	Click "Update" button	Updates successfully and redirect to "Lesson Detail" page. The version of lesson automatically goes up

Alternative Scenario

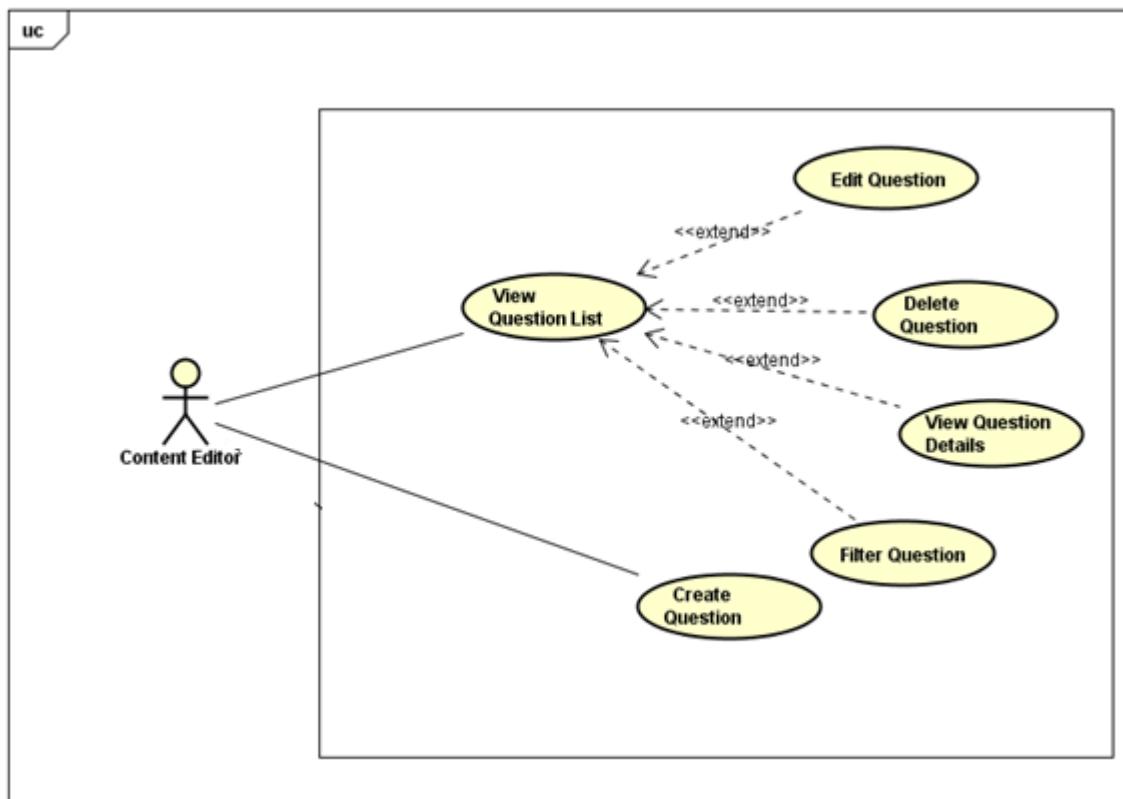
At Step	User	System's Response
3	Clicks "Save draft" button	System saves the draft version of lesson so that the next time user edits the lesson, the draft version can be re-created. The lesson does not make any changes in content.
3	Clicks "Back to Detail" button	System redirects to "Lesson Detail" page
Exception		
At Step	User	System's Response
3	If "Title" textbox is null	Displays the message: "Failed to create new lesson! Please make sure you do not leave any section content & lesson information blank."
3	If "Description" textbox is null	Displays the message: "Failed to create new lesson! Please make sure you do not leave any section content & lesson information blank."
3	If any section of lesson is null or left blank	Displays the message: "Failed to create new lesson! Please make sure you do not leave any section content & lesson information blank."

3.3.3.3.3.6 UC-E-011: Delete Lesson

Use Case Name	Delete Lesson		
Use Case ID	UC-E-011		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/07/2016

Actor	Content Editor	
Description	User deletes lesson	
Pre-condition	<ul style="list-style-type: none"> ● Website is available ● Content editor logged in the system 	
Trigger	<ul style="list-style-type: none"> ● User clicks “Delete” of the lesson in table that user wants to delete, then clicks “Delete” button in “Delete Lesson” popup 	
Post Condition	Delete lesson successfully and system redirects to “Lesson list” page	
Main Success Scenario		
Step	User	System's Response
1	Clicks “Delete” button	Shows “Delete Lesson” popup
2	Clicks “Delete” button	Deletes lesson and closes popup, then reload “Lesson List” page
Alternative Scenario		
At Step	User	System's Response
1	Clicks “Cancel” button in “Delete Lesson” popup	Cancels action of deleting system and closes popup
Exception		
At Step	User	System's Response

3.3.3.3.4 Manage Questions



3.3.3.3.4.1 UC-E-012: View Question List

Use Case Name	View Question List		
Use Case ID	UC-E-012		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	05/08/2016
Actor	Content Editor		
Description	User views list questions		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system 		
Trigger	User clicks "Questions" hyperlink at right-sided bar on dashboard, then clicks "Questions List" hyperlink		
Post Condition	View questions list successfully and system redirects to "Questions List" page		

Main Success Scenario		
Step	User	System's Response
1	Clicks "Questions" hyperlink at right-sided editor menu	Shows "Question List" hyperlink and "Add Question" Hyperlink
2	Clicks "Question List" hyperlink	Redirects to "Question List" page

3.3.3.3.4.2 UC-E-013: Filter question

Use Case Name	Search Question		
Use Case ID	UC-E-013		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	04/08/2016
Actor	Content Editor		
Description	User searches question		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system 		
Trigger	<ul style="list-style-type: none"> • User chooses question type in "Type" dropdown list on "Question List" page • User chooses question level in "Level" dropdown list on "Question List" page • User chooses question skill in "Skill" dropdown list on "Question List" page • User inputs a keyword in "Keyword" textbox • User clicks "Filter" button in question list page 		
Post Condition	Search questions successfully and system filters all questions of the selected combination of type, level, skill and contains the keyword, then displays results		

Main Success Scenario

Step	User	System's Response
1	User chooses question type in "Type" dropdown list	
2	User chooses question level in "Level" dropdown list	

3	User chooses question skill in "Skill" dropdown list	
4	User clicks "Filter" button in question list page	System filters and displays result

3.3.3.3.4.3 UC-E-014: View Question Details

Use Case Name	View Question Details		
Use Case ID	UC-E-014		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	19/07/2016
Actor	Content Editor		
Description	User views question details		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system Done UC-E-014 successfully 		
Trigger	User clicks "Click here to see content" hyperlink of the question in table on "Question List" page		
Post Condition	System redirects to "Question Detail" page		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Click here to see content" hyperlink of a question	Redirect to "Question Detail" page	

3.3.3.3.4.4 UC-E-015: Create Single-typed Question

Use Case Name	Create Single-typed Question		
Use Case ID	UC-E-015		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016

Actor	Content Editor	
Description	User creates a new single-typed question (which includes question content and multiple answers)	
Pre-condition	Website is available Content editor logged in the system	
Trigger	User click “Questions” hyperlink at right-sided editor menu, then click “Add Question” hyperlink	
Post Condition	Creates question successfully and system redirects to “Questions Detail” page If exceptions are thrown, the result must match with output of each described exception.	
Main Success Scenario		
Step	User	System's Response
1	Clicks “Add Question” hyperlink	Redirects to “Add Question” Page
2	Chooses “Single” in “Question Type” dropdown list	
3	Chooses question level in “Level” dropdown list	
4	Chooses question skill in “Test Skill” dropdown list	
5	Inputs question and answer and chooses right answer	
6	Clicks “+” button under answer list	Adds one more answers into answer list of question
7	Adds one more answers into answer list of question	
8	Clicks “X” button at an answer	Removes the answer
9	Clicks “Create” button	Creates question successfully and redirects to “Question Detail” page
Alternative Scenario		
At Step	User	System's Response

4	If user chooses “Listening” in “Test Skill” dropdown list	Displays input that allows user select audio file
Exception		
At Step	User	System's Response
4	If file chosen to upload is not “.mp3” type	Displays the message: “This is not valid audio file. Only .mp3 type is allowed!”
5	If “Question” textbox is null	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer's contents blank.”
5	If “Answer” textbox is null	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer's contents blank.”
5	If user does not choose at least 1 right answer	Displays the message: “There must be at least one right answer in each question!”

3.3.3.3.4.5 UC-E-016: Create Group-typed Question

Use Case Name	Create Group-typed Question		
Use Case ID	UC-E-016		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016
Actor	Content Editor		
Description	User creates new group-typed question (which include a section of reading paragraph and multiple questions)		
Pre-condition	Website is available Content editor logged in the system		
Trigger	User click “Questions” hyperlink at right-sided editor menu, then click “Add Question” hyperlink		
Post Condition	Creates question successfully and system redirects to “Questions Detail” page		

	If exceptions are thrown, the result must match with output of each described exception.	
Main Success Scenario		
Step	User	System's Response
1	Clicks “Add Question” hyperlink	Redirects to “Add Question” Page
2	Chooses “Group” in “Question Type” dropdown list	
3	Chooses question level in “Level” dropdown list	Show level dropdown list
4	Input reading paragraph in “Content” editor	
5	Inputs question and answer and choose right answer	
6	Clicks “+” button under answer list	Adds one more answers into the question’s answer list of question
7	Clicks “+” button under question label	Adds one more single question into question list
8	Clicks “X” button under answer list	Removes the answer from the answer list of the question
9	Clicks “X” button under question label	Remove the question from question list
10	Clicks time icon to select time allowed	
11	Clicks “Create” button	Creates question successfully and redirects to “Question Detail” page
Exception		
At Step	User	System's Response
4	If “Content” section is null or left blank	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer's contents blank.”
5	If “Question” textbox is null	Displays the message: “Failed to create new question! Please make sure

		you do not leave any question & answer's contents blank.”
5	If “Answer” textbox is null	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer's contents blank.”
5	If user does not choose at least 1 right answer in each question	Displays the message: “There must be at least one right answer in each question!”

3.3.3.3.4.6 UC-E-017: Edit Question Single

Use Case Name	Edit Question Single		
Use Case ID	UC-E-017		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	03/08/2016
Actor	Content Editor		
Description	User edits the content of a single-typed question		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system • Done UC-E-014 successfully 		
Trigger	User clicks “Edit” button of a question in table on “Question List” page		
Post Condition	<ul style="list-style-type: none"> • Edit question successfully and redirect to “Question Detail” page • If exceptions are thrown, the result must match with output of each described exception. 		
Main Success Scenario			
Step	User	System's Response	
1	Clicks “Edit” button	Redirects to “Edit Question” page	
2	Chooses question level in “Level” dropdown list		

3	Chooses question skill in "Test Skill" dropdown list	
4	Inputs question and answer and chooses right answer	
5	Clicks "+" button under answer list	Adds one more answers into answer list of question
6	Adds one more answers into answer list of question	
7	Clicks "X" button at an answer	Removes the answer
8	Clicks "Update" button	Updates question successfully and redirects to "Question Detail" page

Alternative Scenario

At Step	User	System's Response
4	If user chooses "Listening" in "Test Skill" dropdown list	Displays input that allows user select audio file
8	Clicks "Back to Detail" button	Cancels action of editing question and redirects to "Question Detail" page

Exception

At Step	User	System's Response
3	If file chosen to upload is not ".mp3" type	Displays the message: "This is not valid audio file. Only .mp3 type is allowed!"
4	If "Question" textbox is null	Displays the message: "Failed to create new question! Please make sure you do not leave any question & answer's contents blank."

4	If "Answer" textbox is null	Displays the message: "Failed to create new question! Please make sure you do not leave any question & answer's contents blank."
4	If user does not choose at least 1 right answer	Displays the message: "There must be at least one right answer in each question!"

3.3.3.3.4.7 UC-E-018: Edit Group-typed Question

Use Case Name	Edit Group-typed Question		
Use Case ID	UC-E-018		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	12/08/2016
Actor	Content Editor		
Description	User edits content of a group-typed question		
Pre-condition	<ul style="list-style-type: none"> Website is available Content editor logged in the system Done UC-E-014 successfully 		
Trigger	User clicks "Edit" button of a question in table on "Question List" page		
Post Condition	<ul style="list-style-type: none"> Edit question successfully and redirect to "Question Detail" page If exceptions are thrown, the result must match with output of each described exception. 		

Main Success Scenario

Step	User	System's Response
1	Clicks "Edit" button	Redirects to "Edit Question" page
2	Chooses question level in "Level" dropdown list	Show level dropdown list
3	Input reading paragraph in "Content" editor	

4	Inputs question and answer and choose right answer	
5	Clicks “+” button under answer list	Adds one more answers into the question’s answer list of question
6	Clicks “+” button under question label	Adds one more single question into question list
7	Clicks “X” button under answer list	Removes the answer from the answer list of the question
8	Clicks “X” button under question label	Remove the question from question list
9	Clicks time icon to select time allowed	
10	Clicks “Update” button	Edits question successfully and redirects to “Question Detail” page

Alternative Scenario

At Step	User	System's Response
9	Click “Back to Detail” button	Cancel edit question and redirect to question detail page

Exception

At Step	User	System's Response
3	If “Content” section is null or left blank	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer’s contents blank.”
4	If “Question” textbox is null	Displays the message: “Failed to create new question! Please make sure you do not leave any question & answer’s contents blank.”

4	If "Answer" textbox is null	Displays the message: "Failed to create new question! Please make sure you do not leave any question & answer's contents blank."
4	If user does not choose at least 1 right answer in each question	Displays the message: "There must be at least one right answer in each question!"

3.3.3.3.4.8 UC-E-019: Delete Question

Use Case Name	Delete Lesson		
Use Case ID	UC-E-019		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	05/07/2016	Date Last Updated	05/08/2016
Actor	Content Editor		
Description	User deletes question		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system • Done UC-E-014 successfully 		
Trigger	User clicks "Delete" button of a question in table on "Question List" page		
Post Condition	Delete question successfully and system reloads "Question List" page		

Main Success Scenario

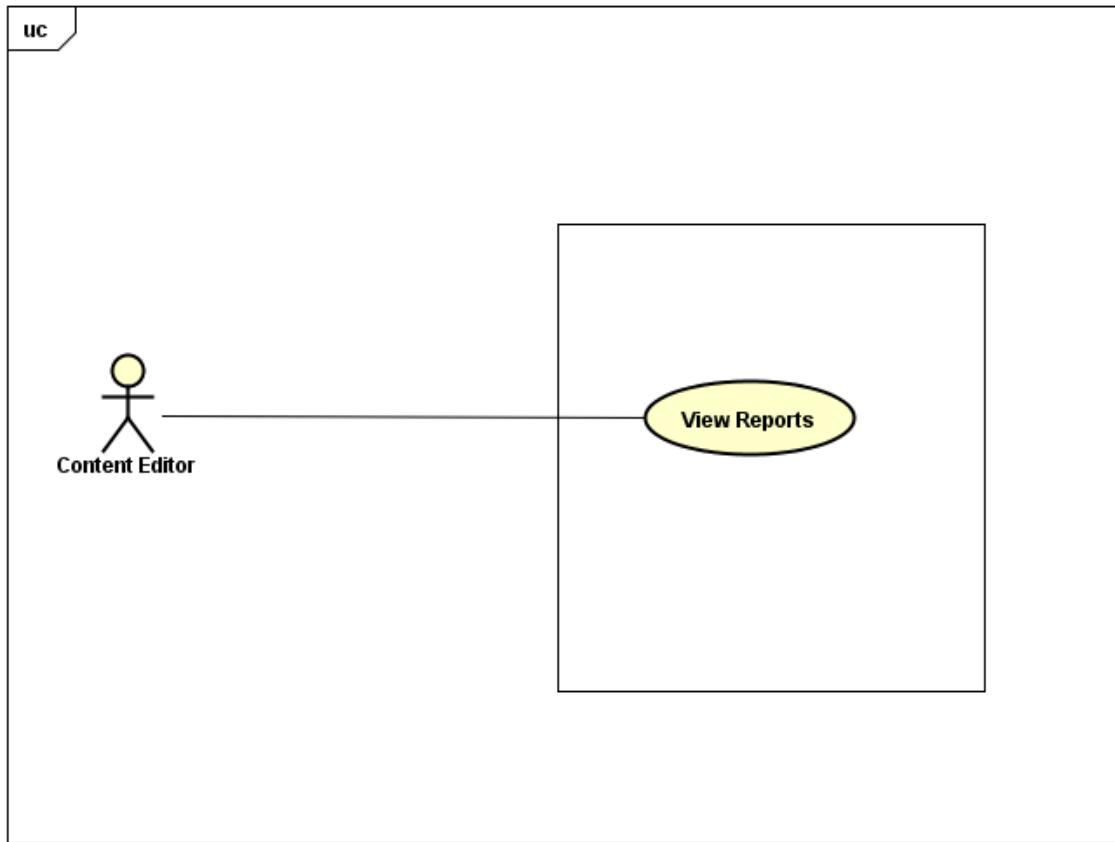
Step	User	System's Response
1	Clicks "Delete" button on "Question List" page	Shows "Delete Question" popup
2	Clicks "Delete" button in popup	Deletes question and closes popup, reload "Question List" page

Alternative Scenario

At Step	User	System's Response
---------	------	-------------------

1	Clicks “Cancel” button in “Delete Question” popup	Cancels action of deleting system and closes popup
Exception		
At Step	User	System's Response

3.3.3.3.5 Manage Reports



3.3.3.3.5.1 UC-E-020: View Reports

Use Case Name	View Reports		
Use Case ID	UC-E-020		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	25/07/2016
Actor	Content Editor		
Description	User views reports about lesson & question sent from user		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system 		

Trigger	User clicks “Reports” hyperlink at the right-sided editor menu		
Post Condition	Redirects to “Report List” page which displays report list		
Main Success Scenario			
Step	User	System’s Response	
1	Click “Reports” hyperlink	Redirects to “Report List” page and display reports list	

3.3.3.3.5.2 UC-E-021: View Reported Lesson

Use Case Name	View Reported Lesson		
Use Case ID	UC-E-021		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	25/07/2016
Actor	Content Editor		
Description	User views the content of the lesson reported by user for review purpose		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system • Done UC-E-020 successfully 		
Trigger	User clicks lesson title hyperlink in a report in table to display the detail of the lesson		
Post Condition	Redirects to “Lesson Detail” page which displays lesson detail		
Main Success Scenario			
Step	User	System’s Response	
1	Clicks lesson title hyperlink	Redirects to “Lesson Detail” page and displays the content of the lesson	

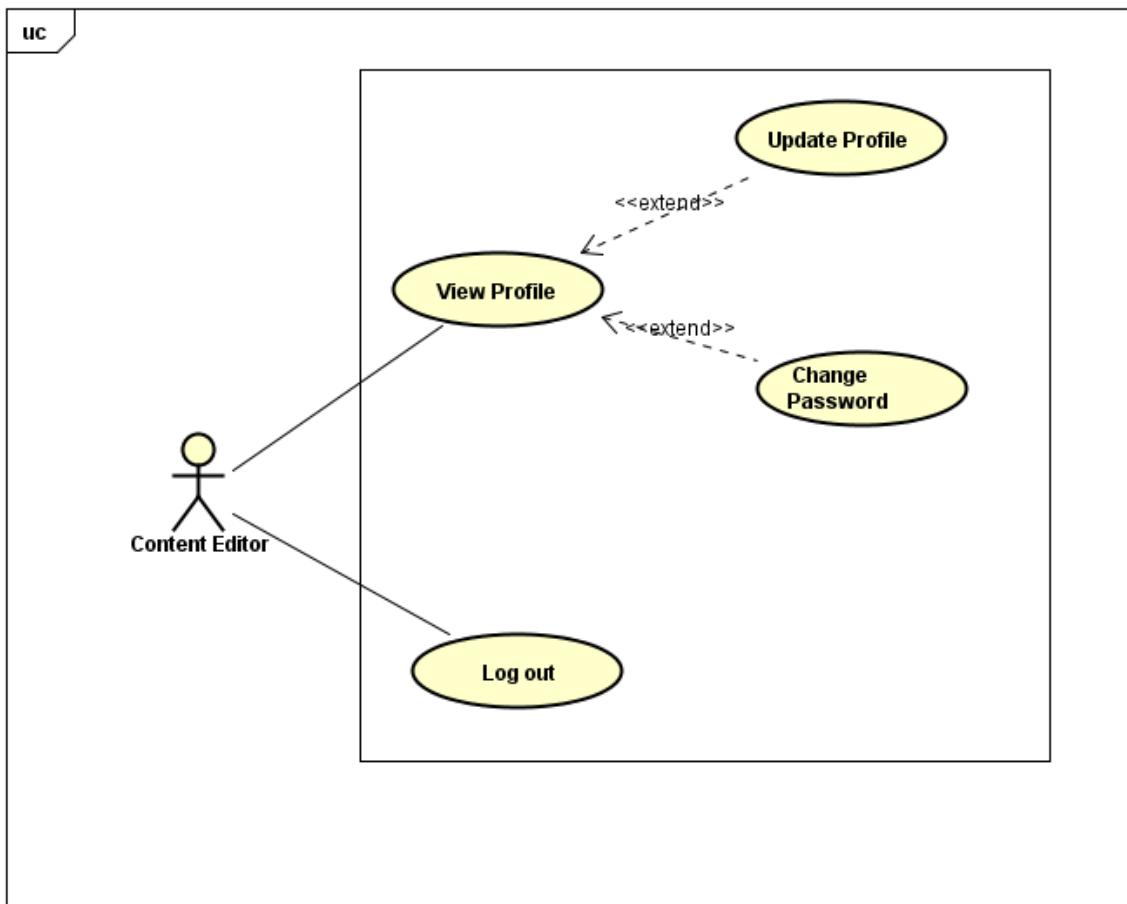
3.3.3.3.5.3 UC-E-021: View Reported Question

Use Case Name	View Reported Question		
Use Case ID	UC-E-021		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	25/07/2016

Actor	Content Editor	
Description	User views the content of the question reported by user for review purpose	
Pre-condition	<ul style="list-style-type: none"> • Website is available • Content editor logged in the system • Done UC-E-020 successfully 	
Trigger	User clicks question ID hyperlink in a report in table to display the detail of the lesson	
Post Condition	Redirects to “Question Detail” page which displays question detail	
Main Success Scenario		
Step	User	System's Response
1	Clicks question ID hyperlink	Redirects to “Question Detail” page and displays the content of the question

3.3.3.4 Administrator

3.3.3.4.1 Manage Account



3.3.3.4.1.1 UC-A-001: View Profile

Use Case Name	View Profile		
Use Case ID	UC-A-001		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Administrator		
Description	User views user's profile detail		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Admin logged in the system 		
Trigger	User clicks username hyperlink at right-sided bar of admin dashboard		
Post Condition	System redirects to "Admin Profile" page		

Main Success Scenario		
Step	User	System's Response
1	Clicks username hyperlink at the right of avatar	Redirects to "Admin Profile" page

3.3.3.4.1.2 UC-A-002: Update Profile

Use Case Name	Update Profile		
Use Case ID	UC-A-002		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	04/07/2016	Date Last Updated	29/07/2016
Actor	Administrator		
Description	User edits information of their own profile		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Admin logged in the system • Done UC-A-001 successfully 		
Trigger	<ul style="list-style-type: none"> • User inputs information (including first name, last name, birthday, address, hobbies, about me, favorite quote, website) • User clicks "Save" button at the bottom of "Update Profile Information" popup Information" popup 		
Post Condition	<ul style="list-style-type: none"> • System closes the dialog, reloads "Admin Profile" page and shows update successfully • If exceptions are thrown, the result must match with output of each described exception. 		

Main Success Scenario

Step	User	System's Response
1	Clicks "Update Profile" in "Actions" dropdown list	Shows "Update Profile Information" popup
2	Inputs first name, last name, birthday, address, hobbies, about me, favorite quote, website in respective textbox	

3	Clicks “Save” button	Updates profile successfully, closes dialog and reload “Admin Profile” page
Alternative Scenario		
At step	User	System’s Response
3	Clicks “Cancel” button	Cancels editing profile action and closes “Update Profile Information” popup
Exception		
At Step	User	System’s Response
2	If any of those textbox is null	System skips and does not update the information of the textbox

3.3.3.4.1.3 UC-A-004: Change Password

Use Case Name	Change Password		
Use Case ID	UC-A-004		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Administrator		
Description	User change their password		
Pre-condition	<ul style="list-style-type: none"> Website is available Done UC-A-001 successfully Admin logged in the system 		
Trigger	User clicks “Change password” hyperlink in “Actions” dropdown list on “Editor Profile” page		
Post Condition	<ul style="list-style-type: none"> User click “Change Password” in “Actions” dropdown list User inputs current password, new password and re-type new password User clicks “Change” button at the bottom of “Change Password” popup 		
Main Success Scenario			

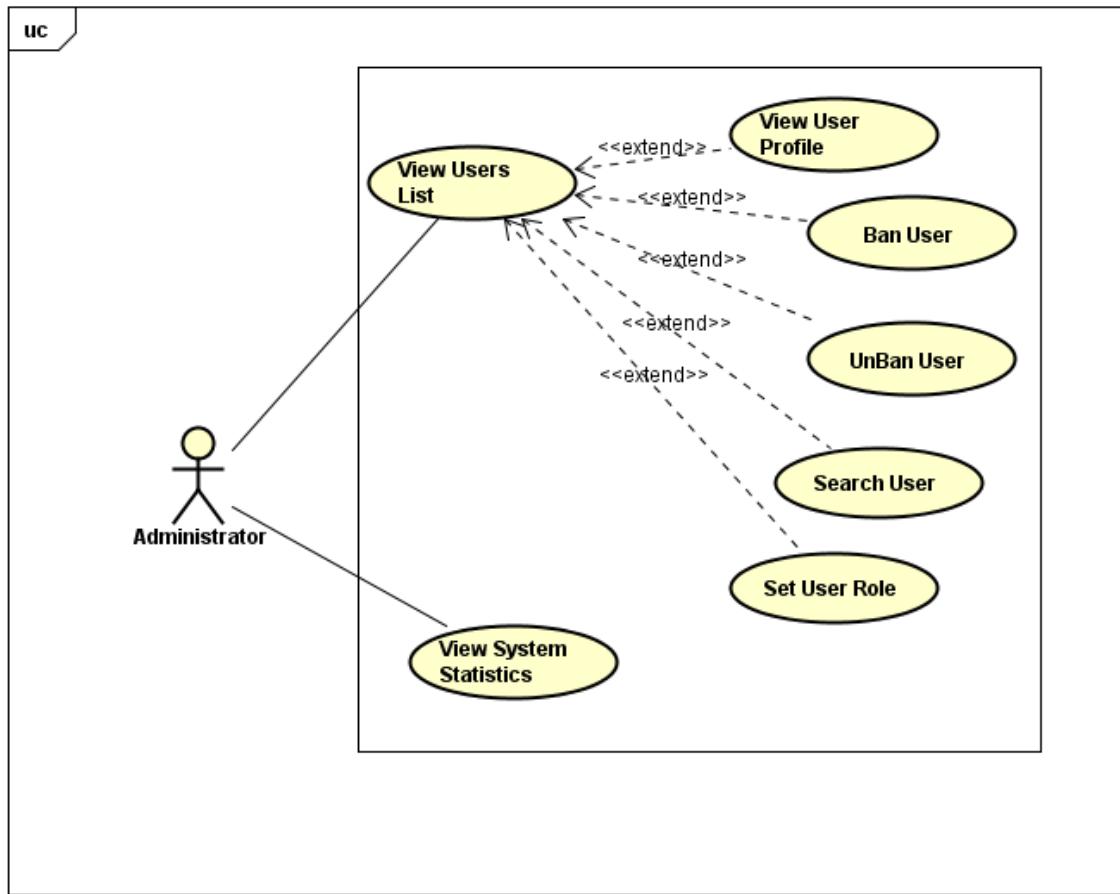
Step	User	System's Response
1	Clicks "Change Password" in "Actions" dropdown list.	Shows "Change Password" popup
2	Inputs current password, new password and password confirmation	
3	Click "Save" button.	Changes password successfully and closes popup
Alternative Scenario		
At Step	User	System's Response
1	Clicks "Cancel" button.	Cancels changing password action and closes "Change Password" popup
Exception		
At Step	User	System's Response
2	If "Current Password" textbox is null	Displays the message: "Password required!"
2	If "New Password" textbox is null	Displays the message: "Password required!"
2	If new password and re-typed password is not the same	Displays the message: "Password does not match"
2	If user inputs new password that is less than 6 characters	Displays the message: "Password must be at least 6 characters long!"
2	If user inputs new password that is more than 32 characters	Displays the message: "Password must be no more than 32 characters long!"
2	If current password is incorrect	Displays the message: "Incorrect password!"

3.3.3.4.1.4 UC-A-005: Logout

Use Case Name	Logout

Use Case ID	UC-A-003		
Created By	QuanNH	Last Updated By	ThangPD
Date Created	27/06/2016	Date Last Updated	19/07/2016
Actor	Administrator		
Description	User logs out of the Veazy system		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Admin login the website 		
Trigger	User clicks “Logout” hyperlink at right-sided bar of admin dashboard		
Post Condition	Signs user out successfully and redirects to “Login” page		
Main Success Scenario			
Step	User	System's Response	
1	Click “Logout” hyperlink at admin dashboard	System redirects to “Login” page	

3.3.3.4.2 Manage User



3.3.3.4.2.1 UC-A-006: View User List

Use Case Name	View User List		
Use Case ID	UC-A-006		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User views list of Veazy members		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Admin logged in the system 		

Trigger	Admin click “User Management” hyperlink at the right-sided administration menu	
Post Condition	System redirects to “User List” page which displays list of Veazy members, including both User and Editor	
Main Success Scenario		
Step	User	System's Response
1	Clicks “User Management” hyperlink	Redirects to “User List” page and display member list

3.3.3.4.2.2 UC-A-007: View User Profile

Use Case Name	View User Profile		
Use Case ID	UC-A-007		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User views member's profile		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-A-006 successfully • Admin logged in the system 		
Trigger	Users username hyperlink of member in table on “User List” page		
Post Condition	System shows “User Profile” popup which displays user information including occupation, first name, last name, birthday, address, hobbies, favorite quote and website.		
Main Success Scenario			
Step	User	System's Response	
1	Clicks username hyperlink	Shows “User Profile” dialog and display user information	

3.3.3.4.2.3 UC-A-008: Assign User as Admin

Use Case Name	Assign User as Admin		
Use Case ID	UC-A-008		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User assigns a user as admin of Veazy		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-A-006 successfully • Admin logged in the system 		
Trigger	User clicks “Manage” hyperlink of a user in table, then clicks “Assign as admin” in dropdown list		
Post Condition	The user is assigned as new admin of Veazysystem and can access exclusive features of admin after login.		
Main Success Scenario			
Step	User	System's Response	
1	Click “Manage” hyperlink	Show dropdown list that contains “Assign as editor”, “Assign as admin”, “Ban” hyperlink	
2	Clicks “Assign as admin”	Changes user type of the user to “Admin” and reloads “User List” page, shows update successfully	
Alternative Scenario			
At Step	User	System's Response	
Exception			
At Step	User	System's Response	

3.3.3.4.2.4 UC-A-009: Assign User as Editor

Use Case Name	Assign User as Editor		
Use Case ID	UC-A-009		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User sets role for member of system		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-A-006 successfully • Admin logged in the system 		
Trigger	User clicks “Manage” hyperlink of a user in table, then clicks “Assign as editor” in dropdown list		
Post Condition	The user is assigned as new editor of Veazy system and can access exclusive features of editor after login.		
Main Success Scenario			
Step	User	System's Response	
1	Click “Manage” hyperlink	Show dropdown list that contains “Assign as editor”, “Assign as admin”, “Ban” hyperlink	
2	Clicks “Assign as editor”	Changes user type of the user to “Editor” and reloads “User List” page, shows update successfully	
Alternative Scenario			
At Step	User	System's Response	
Exception			
At Step	User	System's Response	

3.3.3.4.2.5 UC-A-010: Ban User

Use Case Name	Ban user		
Use Case ID	UC-A-010		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User bans member		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-A-005 successfully • Admin logged in the system • The status of the user is "Active" 		
Trigger	User clicks "Manage" hyperlink of a user in table, then clicks "Ban" in dropdown list		
Post Condition	Systems reload "User List" page and shows update successfully. The user can no longer log in Veazy system under their username		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Manage" hyperlink	Shows dropdown list that contains "Assign as editor", "Assign as admin", "Ban" hyperlink	
2	Clicks "Ban"	Changes user's status to "Banned" and prevents that user from logging into system.	
Alternative Scenario			
At Step	User	System's Response	
Exception			
At Step	User	System's Response	

3.3.3.4.2.6 UC-A-011: Unban User

Use Case Name	Unban user		
Use Case ID	UC-A-011		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User unban member		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Done UC-A-006 successfully • Admin logged in the system • The status of the user is "Banned" 		
Trigger	User clicks "Manage" hyperlink of a user in table, then clicks "Unban" in dropdown list		
Post Condition	Systems reload "User List" page and shows update successfully. The user is enabled to log in Veazy system under their username		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Manage" hyperlink	Shows dropdown list that contains "Assign as editor", "Assign as admin", "Unban" hyperlink	
2	Clicks "Unban"	Changes user's status to "Active" and allow that user to log into system.	
Alternative Scenario			
At Step	User	System's Response	
Exception			
At Step	User	System's Response	

3.3.3.4.2.7 UC-A-012: View System Statistics

Use Case Name	View System Statistics		
Use Case ID	UC-A-011		
Created By	QuanNH	Last Updated By	QuanNH
Date Created	27/06/2016	Date Last Updated	12/07/2016
Actor	Administrator		
Description	User views statistics of system		
Pre-condition	<ul style="list-style-type: none"> • Website is available • Admin logged in the system 		
Trigger	Clicks "Statistics" hyperlink at right-sided admin menu		
Post Condition	Systems redirect to "System Statistics" page which displays the statistics of users, lessons, questions and reports		
Main Success Scenario			
Step	User	System's Response	
1	Clicks "Statistics" hyperlink	Redirects to "System Statistics" page and displays statistics of users, lessons, questions and reports	

3.3.4 Non-Functional Specification Requirements

3.3.4.1 Usability

- Interface should be simple, clear, user-friendly, easy to learn how to use and easy to remember how to use.
- Interface design should support the following from the perspectives of its primary users:
 - Efficiency of use: each action is taken by no more than 8 steps.
 - Intuitiveness: the interface is easy to learn and navigate: buttons, headings, and help/error messages are simple to read and understand.
 - Low perceived workload: the interface appears easy to use, rather than intimidating, demanding and frustrating.
- Users need to understand the function of each section to be able to use reasonable discretion and its effectiveness.
- Users should be supported and guided how to use the functionality of the system to use system easily after short time.
 - In the role “Administrator” – after 2 hours of training
 - In the role “Content Editor” – after 1 hour of training
 - In the role “Member” – after 30 minutes of training
 - In the role “Guest” – after 15 minutes of training
- The system should have the flexibility to be can satisfied future changed as:
 - Can add more lessons, tests and new words.
 - Changes in processing of information.
- The system must meet the requirements for data memory, need to calculate the amount of memory to be able to meet the requirements of the development of the system.

3.3.4.2 Security

- Password of user is md5-hashed before being sent in request to server and saved into database
- Request sent from client is authorize using cookies.

3.3.4.3 Maintainability & Extensibility

- Strictly follow Java & JavaScript best practices and English-like syntax naming convention in order to help improve readability of source code and keep the Veazy website maintainable in the future.

3.3.4.4 Performance

- Response time when accessing an action on the system depending on system condition such as number of visitors, access tasks.
- Time for loading image, media will be at most 15 seconds and time to progress any function will not exceed 17 seconds.

3.3.4.5 Supportability

- Web system is written in Java and Bootstrap, easily manage and upgrade and easily connect to database system.

3.3.4.6 Design Constraints

- Because the system is built on the Web Platform, therefore development language easily manages and upgrade.
- The system must execute independently, less influenced by other systems and capable of performing well on many different operating systems.

3.3.4.7 User Manual Documentation

- Integration guild with both English and Japanese language, image and clear explanation

4 SOFTWARE DESIGN DESCRIPTION

4.1 SDD Overview

This document provides a comprehensive architectural overview of the system VEAZY, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions that have been made on the system.

This section of document is the **software design description** of the project in order to develop the software, it includes four main parts: Architecture Design, Class Design, Detail Design, and Database design.

4.2 Architecture Design

4.2.1 MVC Model

The architecture of Veazy system is structured based on MVC model combined with layered architecture, Spring framework and AngularJS framework.

4.2.1.1 MVC Model overview

Model-view-controller (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts: the model, the view, and the controller, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The model consists of application data, business rules, logic and functions. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as bar chart for management and a tabular view for accountants. The controller mediates input, converting it to commands for the model or view.

The MVC design pattern assigns objects in an application one of three roles: model, view, or controller. The pattern defines not only the roles objects playing in the application but the way objects communicate with each other as well. Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries. The collection of objects of a certain MVC type in an application sometimes referred to as a layer—for example, model layer.

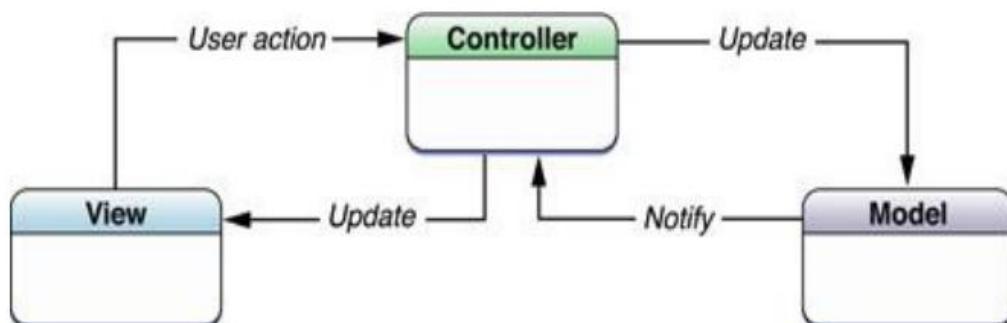


Figure 4-1: MVC Model

In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them:

- **A controller:** can send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document). It can also send commands to the model to update the model's state (e.g., editing a document).
- **A model:** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.
- **A view:** requests from the model the information that it needs to generate an output representation to the user.

4.2.1.2 Advantages and disadvantages of MVC model

- Advantages:
 - The MVC model demonstrates professionalism in programming and design analysis. It is divided into independent components to help develop applications faster, simpler, easier upgrades and maintenance.
 - Many MVC vendor framework tool kits are available.
 - Multiple views synchronized with same data model.
 - Easy to change or plug in new interface views, allowing updating of interface views with new technologies without overhauling the rest of system.
 - Very effective for deployment if graphic, programming and database development professionals are working in a team in a designed project.
- Disadvantages:
 - For small projects that apply MVC model caused cumbersome, time consuming in development process.
 - Time consuming to transits data between components.
 - Not suitable for agent-oriented applications such as interactive mobile and robotics applications.
 - Multiple pairs of controllers and views based on the same data model make data model change expensive.
 - The division between the View and the Controller is not clear in some cases.

4.2.1.3 The reason of choosing MVC model

- Easily manage the complexity of application by dividing the application into three components: model, view and controller.

- Better support for TDD.
- It is a good support for application built by project team that has many developers and designers but still managed application features.
- Veazy system is an incomplete system for now. Veazy is built in the way that towards extensibility and maintainability in the future.

4.2.2 SpringMVC framework

4.2.2.1 Overview

Spring's web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central Servlet that dispatches requests to controllers and offers other functionality that facilitates the development of web applications. Spring's DispatcherServlet however, does more than just that. It is completely integrated with the Spring IoC container and as such allows you to use every other feature that Spring has.

The Spring's web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The Model encapsulates the application data and in general they will consist of POJO.
- The View is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The Controller is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

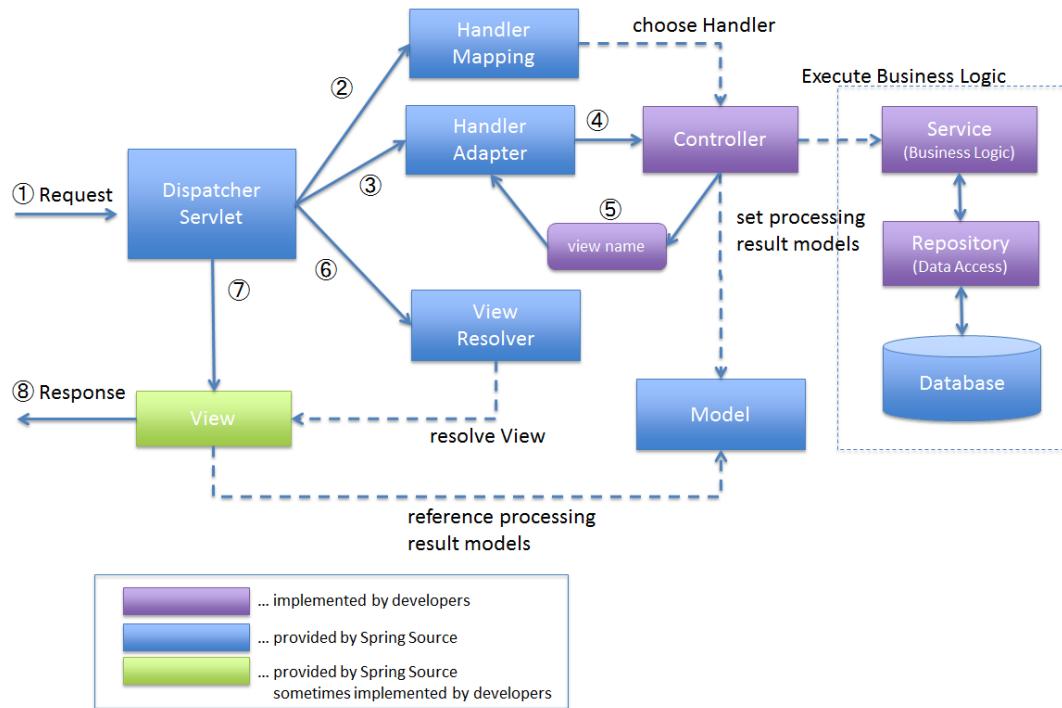


Figure 4-1 SpringMVC process flow

1. *DispatcherServlet* receives the request.
2. *DispatcherServlet* dispatches the task of selecting an appropriate controller to *HandlerMapping*. *HandlerMapping* selects the controller which is mapped to the incoming request URL and returns the (selected *Handler*) and *Controller* to *DispatcherServlet*.
3. *DispatcherServlet* dispatches the task of executing of business logic of *Controller* to *HandlerAdapter*.
4. *HandlerAdapter* calls the business logic process of *Controller*.
5. *Controller* executes the business logic, sets the processing result in *Model* and returns the logical name of view to *HandlerAdapter*.
6. *DispatcherServlet* dispatches the task of resolving the *View* corresponding to the *View* name to *ViewResolver*. *ViewResolver* returns the *View* mapped to *View* name.
7. *DispatcherServlet* dispatches the rendering process to returned *View*.
8. *View* renders *Model* data and returns the response.

4.2.2.2 Advantages and disadvantages

- Advantages
 - No need to reinvent the wheel: It utilizes some of the well-known technologies, ORM frameworks, logging frameworks, JEE, JDK timers, Quartz and so on
 - Dependency Injection gears up testability.
 - Inversion control and APIs: Spring framework provides inversion control and APIs to translate technology-driven exceptions, specifically thrown by JDBC, Hibernate or JDO, into unchecked and consistent ones.
 - Modularity: Spring makes it easy for the developers to know which packages or classes are to be used and which one should be ignored.
 - Consistent Transaction Management: With the help of consistent transaction management interface, Spring framework easily scale down or scale up local as well as global transactions.
 - Spring AOP also brings tons of benefits: There is no need for a developer to have a separate compilation unit or a separate class loader.
- Disadvantages
 - Complex: One of the major criticisms faced by the Spring framework is that it is complex.
 - Longer Learning Curve: it would be quite difficult to learn Spring framework. The main reason behind this is a whole host of new programming methods and detailing require understanding how to set up the Spring XML configuration file.
 - All the Spring applications require a lot of XML: the applications developed using Spring framework often require a huge amount of XML
 - Tons of parallel mechanisms frustrate developers: It makes developers to spend lots of understanding each of them and choose the best one among them.
 - Lack of Guidelines: No clear guidance on cross-site scripting attacks and cross-site request forgery attacks in Spring MVC documentation

4.2.2.3 The reason of choosing

- - Spring provides a very clean division between controllers, JavaBean models, and views.
- Spring's MVC is very flexible, it is entirely based on interfaces.

- Spring provides interceptors as well as controllers, making it easy to factor out behavior common to the handling of many requests.
- Spring Controllers are configured via IoC. This makes them easy to test, and beautifully integrated with other objects managed by Spring.
- No ActionForms. Bind directly to domain objects.
- Spring has a well defined interface to business layer Architectural Representation

4.2.3 HibernateORM framework

4.2.3.1 Overview

Hibernate is an implementation of the Java persistence API (JPA) specification, it can easily used in any environment supporting JPA including Java SE applications, Java EE applications, Java EE application servers, Enterprise OSGi containers and so on. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieve the developer from 95% of common data persistence related programming tasks.

Hibernate sits between traditional Java objects and database server to handle all the work in persisting those objects based on the appropriate O/R mechanisms and patterns.



Figure 4-2 Hibernate ORM

4.2.3.2 Advantages & Disadvantages

- Advantages
 - Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
 - Provides simple APIs for storing and retrieving Java objects directly to and from the database.
 - If there is change in Database or in any table then the only need to change XML file properties.

- Abstract away the unfamiliar SQL types and provide us to work around familiar Java Objects.
 - Hibernate does not require an application server to operate.
 - Manipulates complex associations of objects of your database.
 - Minimize database access with smart fetching strategies.
 - Provides simple querying of data.
- Disadvantages
 - Slower than JDBC: Hibernate is slower than pure JDBC as it is generating lots of SQL statements in runtime.
 - Not suitable for Batch processing: It is advisable to use pure JDBC for batch processing.
 - Not suitable for Small projects: For small project having few tables, it is useless to work with hibernate.
 - Does not allow multiple inserts: Hibernate does not allow some type of queries which are supported by JDBC.
 - Generates complex queries with many joins: For complex data, mapping from Object-to-tables and vice versa reduces performance and increases time of conversion.

4.2.3.3 The reason of choosing

- Productivity: It helps developers get rid of writing complex and tedious SQL statement, no more need of JDBC APIs for result set or data handling. It makes developers more concentrate on the business logic and increase the project's productivity.
- Maintainability: It helps reduce the lines of code, makes system more understandable and emphasizes more on business logic rather than persistence work (SQLs).
- Portability: It abstracts our application away from the underlying SQL database and sql dialect. Switching to other SQL database requires few changes in Hibernate configuration file.

4.2.4 AngularJS framework

4.2.4.1 Overview

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

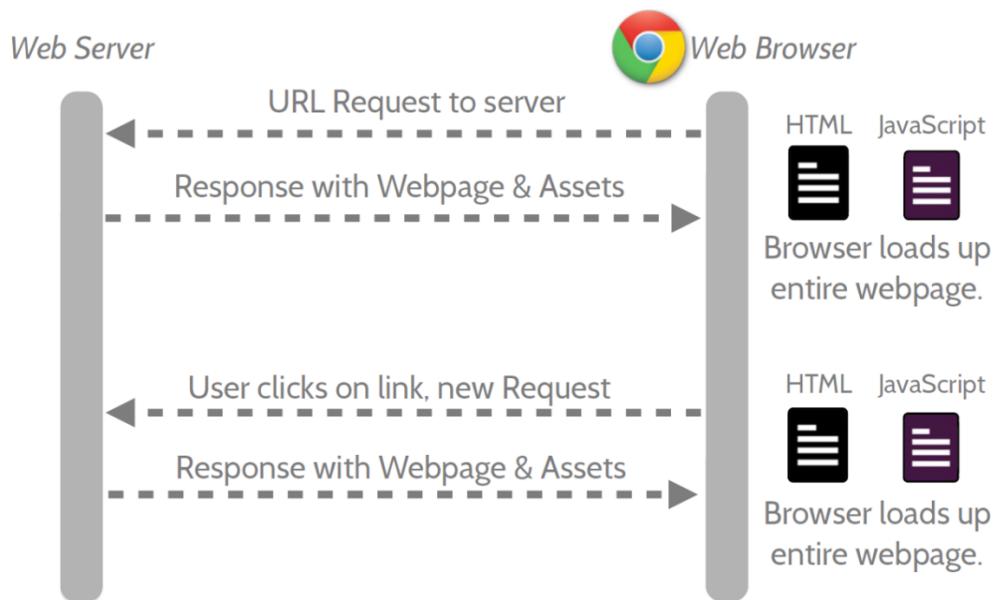


Figure 4-3 Multi-page flow

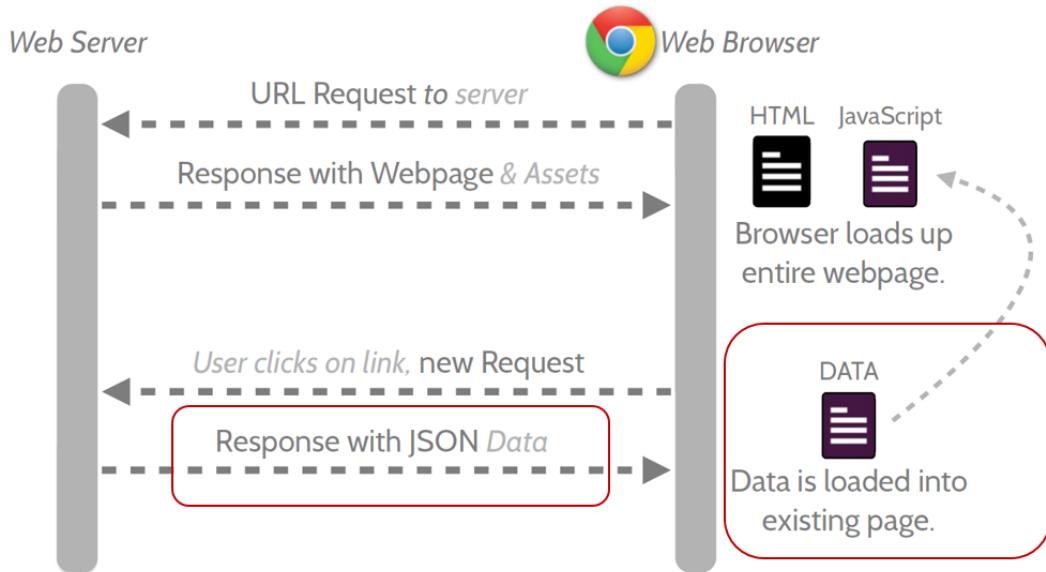


Figure 4-4 Single-page flow

4.2.4.2 Advantages & Disadvantages

- Advantages
 - AngularJS provides capability to create Single Page Application in a very clean and maintainable way.
 - AngularJS provides data binding capability to HTML thus giving user a rich and responsive experience
 - AngularJS code is unit testable.
 - AngularJS uses dependency injection and make use of separation of concerns.
 - AngularJS provides reusable components.
 - With AngularJS, developer write less code and get more functionality.
 - In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.
- Disadvantages
 - Not Secure – Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.
 - Not degradable – If your application user disables JavaScript then user will just see the basic page and nothing more.

4.2.4.3 The reason of choosing

- Expressive User Interface: Angular employs HTML to define application user interface. HTML carries special attributes explaining which controllers to use for elements.
- Less code: By ignoring writing of own pipeline, use of html, simple data model and use of filters, angular makes it coding free for developers.
- Testability: AngularJs has been developed keeping in mind the testing process of coding and developing at each and every stage of application development so as to avoid wastage of time.

4.2.5 Architectural Representation

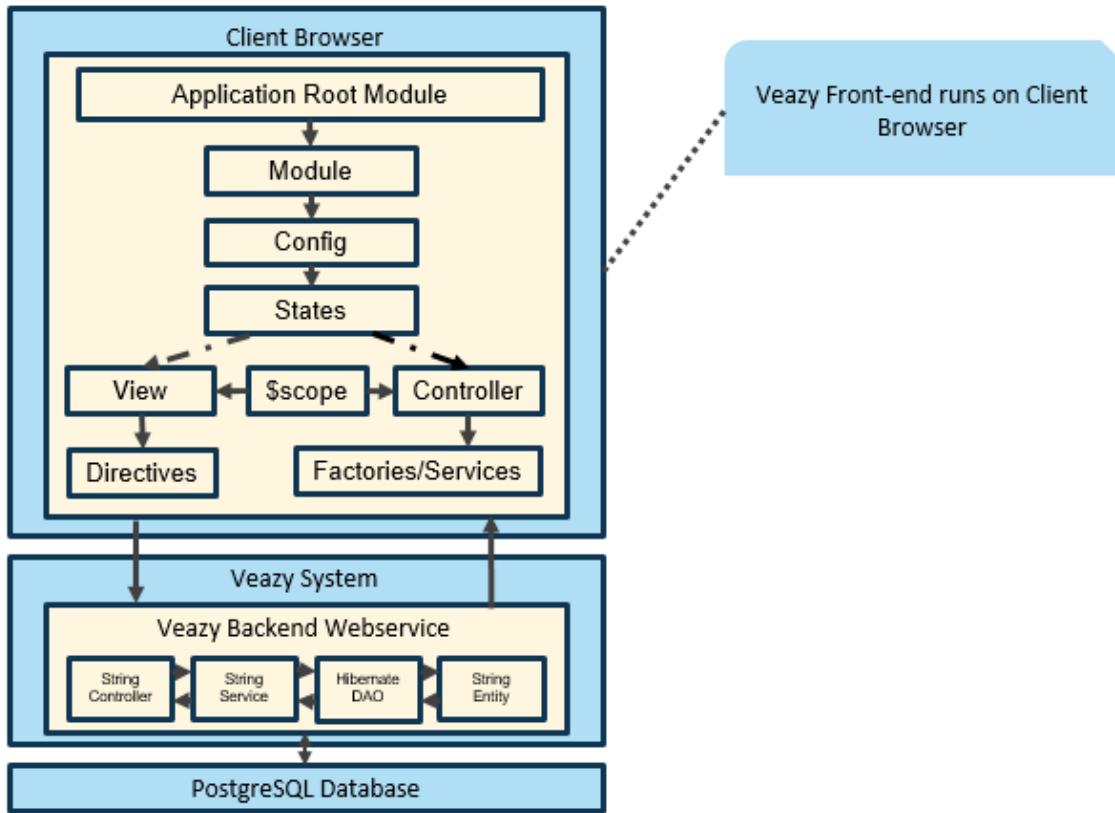


Figure 4-2: System overview

We follow MVC architecture to implement the Veazy project. MVC offers architectural benefits over jQuery and AngularJS — it helps us write better-organized and therefore more maintainable code.

Model is where the application's data objects are stored. A model object is in charge of encapsulating application state and one object could be related to other objects establishing a one-to-one or one-to-many relationship. When a model changes, typically it will notify its observers that a change has occurred. As with any data object it contains instance variables and getter/setter methods.

DAO is an abbreviation of Data Access Object which provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, DAO provide some specific data operations without exposing details of the database.

Service defines set of available operations in regards with interfacing client layers i.e. encapsulates the application's business logic.

Controller is the decision maker and the glue between the model and view; it handles user actions and gestures, and responds to user events.

4.2.6 Architecture Goals and Constraints

Availability: The application must be available 95% of time. Users can access to it everywhere from there .Web browser with internet connection.

Maintainability:

- Coding standards and naming conventions
 - Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
 - All code must be clearly commented, including class, method documentations.
 - If some components are reused, the documentations of those components must also be included.
- Design
 - The design of the system must be loosely coupled that changes on some module will not affect others.
- Logging
 - All the errors should be logged, supporting for bug fixing and maintenance.
 - All strange or sensitive situations should also be logged.

Usability: Usability Requirements support the following from the perspective of its primary users:

- Efficiency of use: user can complete each function in less than 12 actions
- Intuitiveness: all help/error messages are simple to understand; user can know exactly how to do each feature after one time using it.

Capacity and scalability: throughput, storage and growth requirements.

4.2.7 Use-Case View

The use-case view consists of 4 parts according to 4 user roles in the system: guest, member, editor and admin.

4.2.7.1 Guest

4.2.7.1.1 Register account

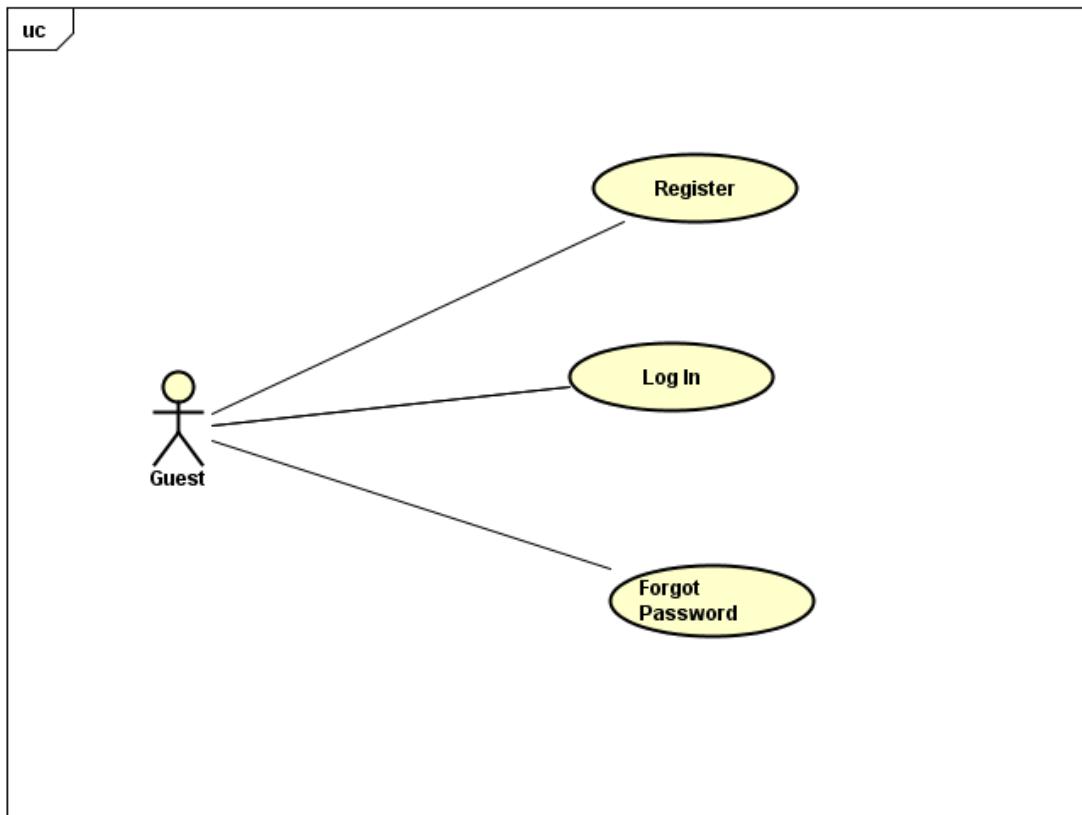


Figure 4-5 Register account

No	Use-case name	Actor	Description
1	Register	Guest	Registering new account to log into the system
2	Log In	Guest	Logging into the system
3	Forgot Password	Guest	Resetting user's password

4.2.7.1.2 Learning

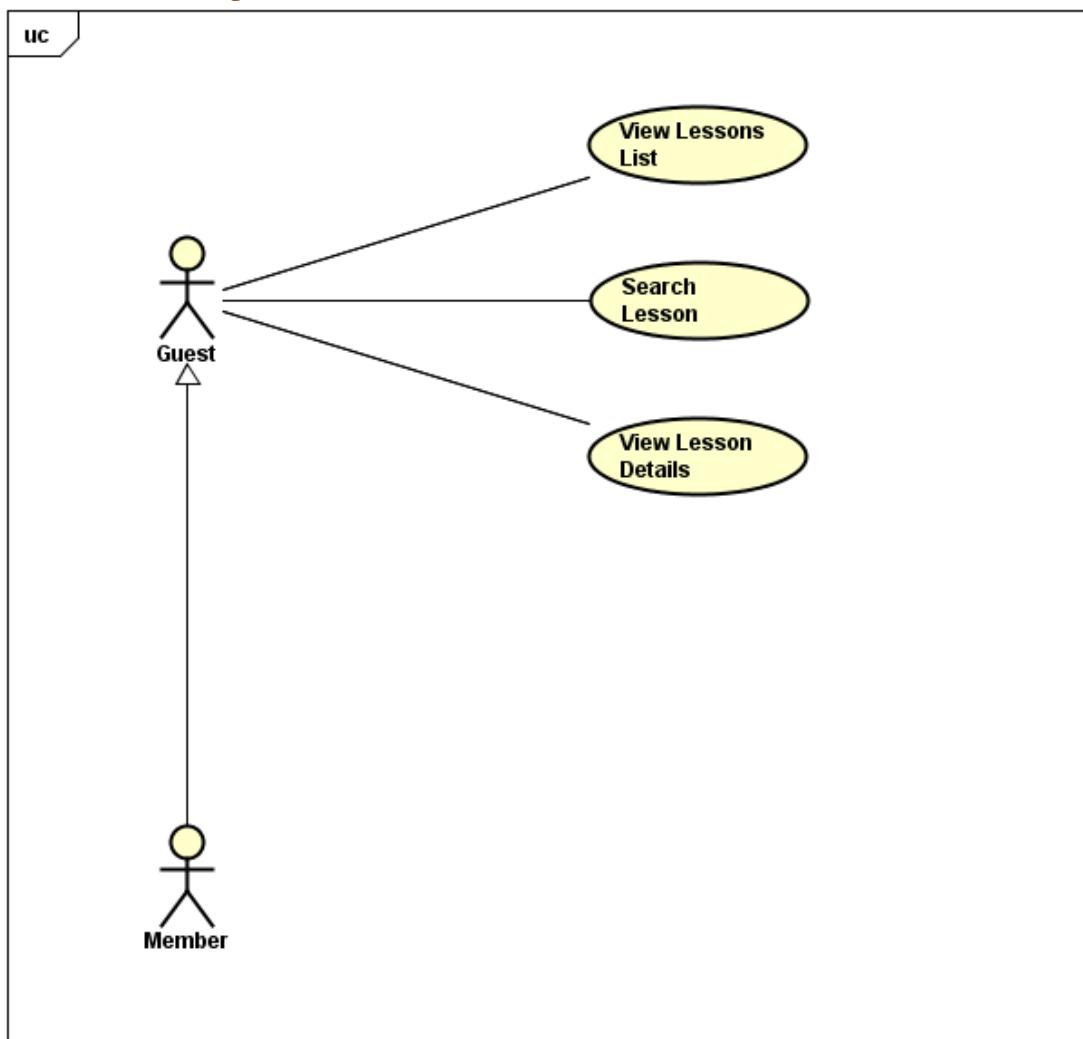


Figure 4-6 Learning

No.	Use-case name	Actor	Description
1	View Lesson List	Guest, Member	Viewing lesson list of courses
2	Search Lesson	Guest, Member	Searching lesson
3	View Lesson Details	Guest, Member	Viewing detail of lesson

4.2.7.1.3 Testing

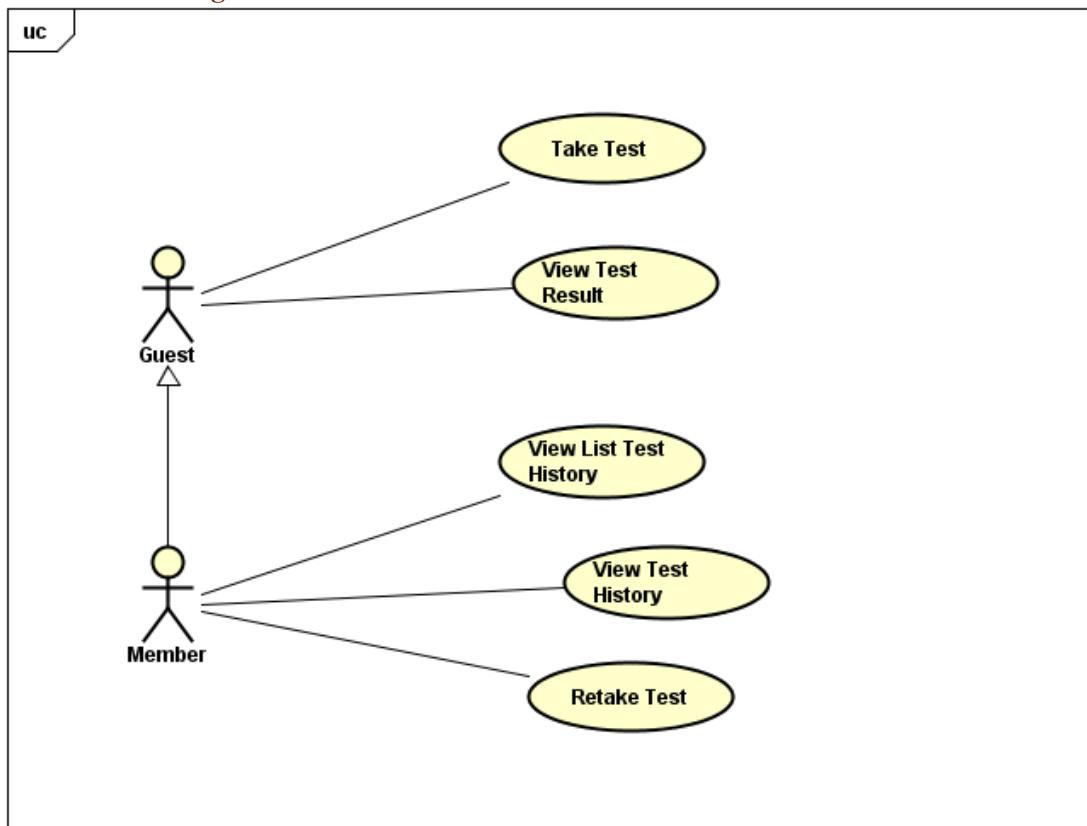


Figure 4-7 Testing Use Case

No.	Use-case name	Actor	Description
1	Take Test	Guest, Member	Taking a multiple-choice test
2	View Test Result	Guest, Member	Viewing result of test

4.2.7.2 Member

4.2.7.2.1 Private Account Management

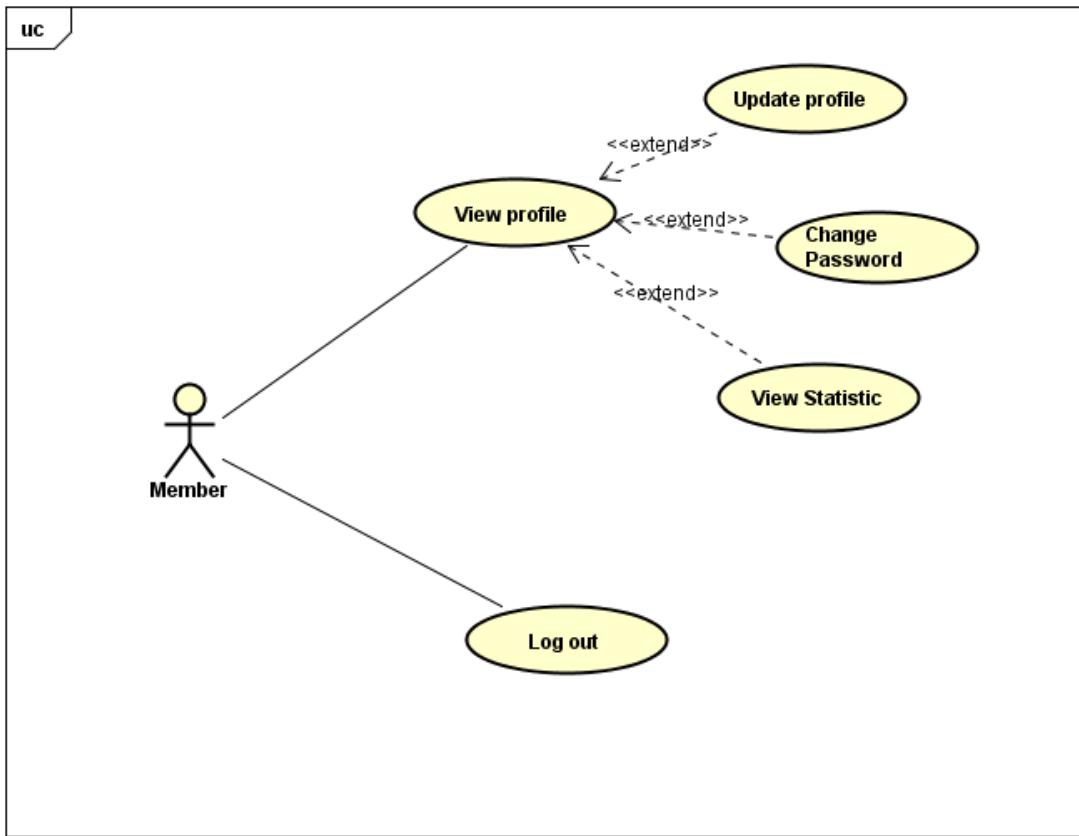


Figure 4-8 Private Account Management

No.	Use-case name	Actor	Description
1	View Profile	Member	Viewing detail information of their profile
2	Update Profile	Member	Updating user's profile
3	Change Password	Member	Changing user's password
4	View Statistic	Member	Viewing statistics of user
5	Log Out	Member	Logging out of system

4.2.7.2.2 Testing

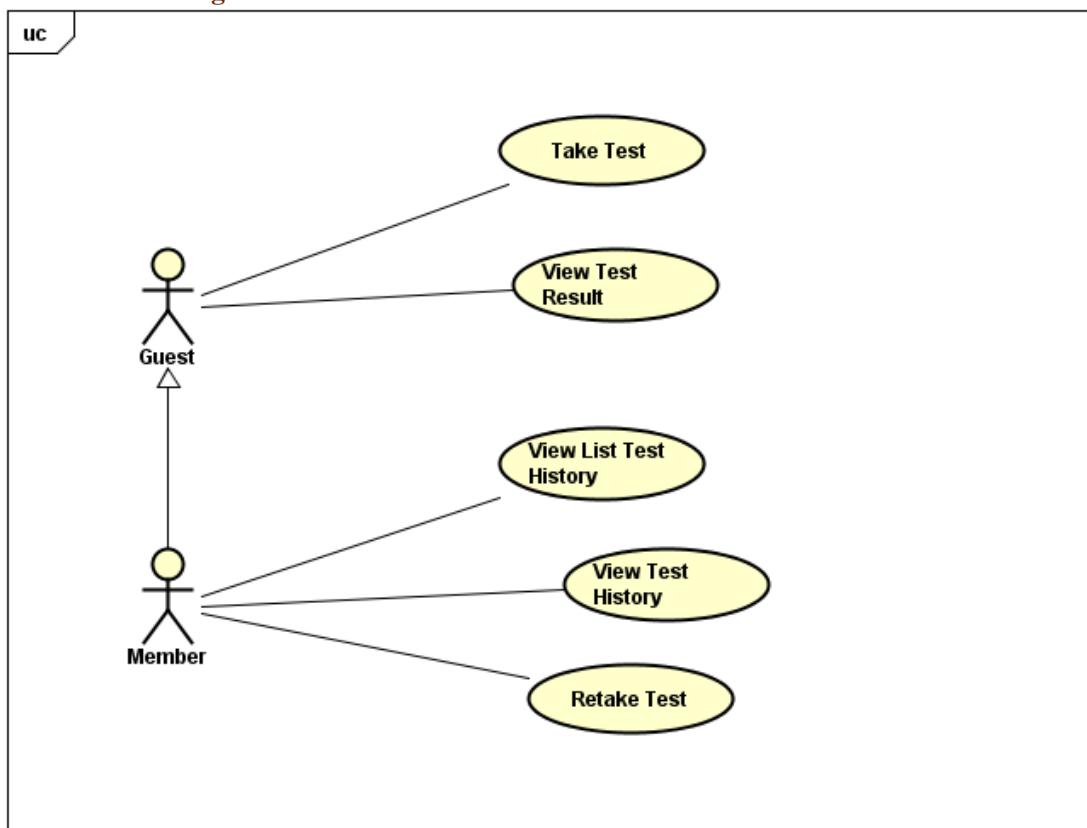


Figure 4-9 Testing

No.	Use-case name	Actor	Description
1	View List Test History	Member	Viewing list of test history
2	View Test History	Member	Viewing detail of test history
3	Retake Test	Member	Retaking a test in list test history

4.2.7.2.3 Content Reporting

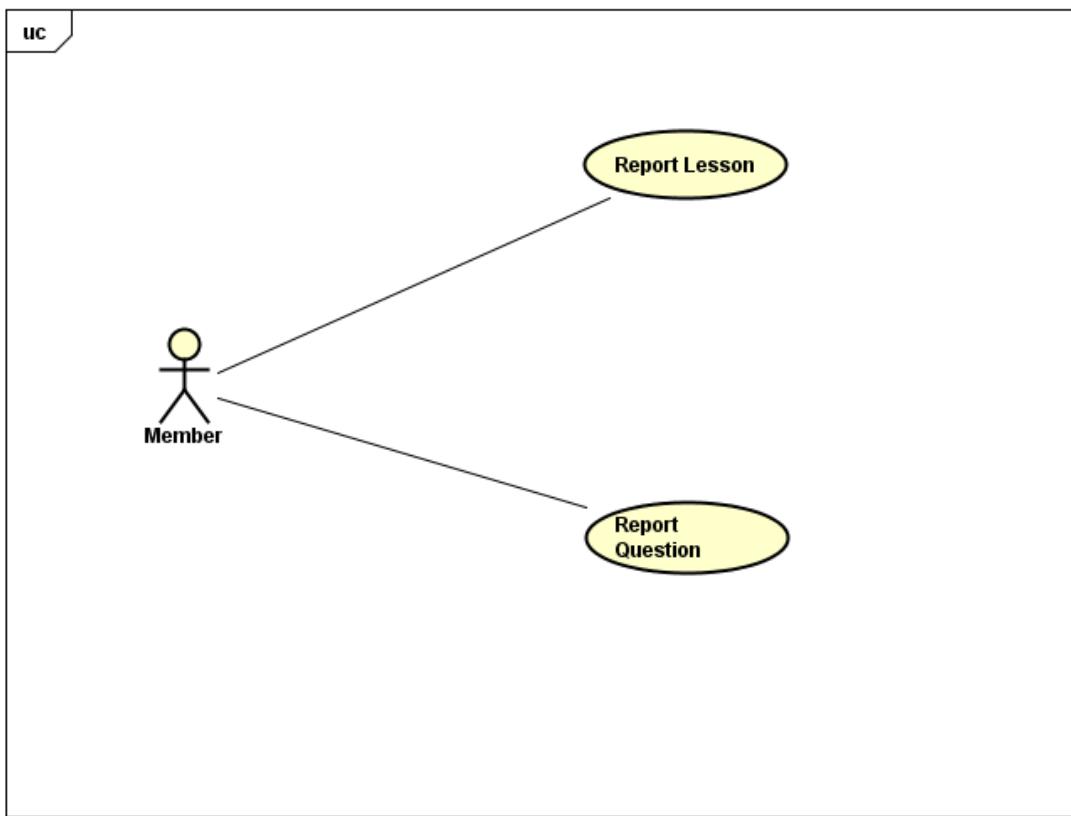


Figure 4-10 Content Reporting

No.	Use-case name	Actor	Description
1	Report Lesson	Member	Reporting content of lessons that having wrong content
2	Report Question	Member	Reporting content of questions that having wrong content

4.2.7.3 Editor

4.2.7.3.1 Private Account Management

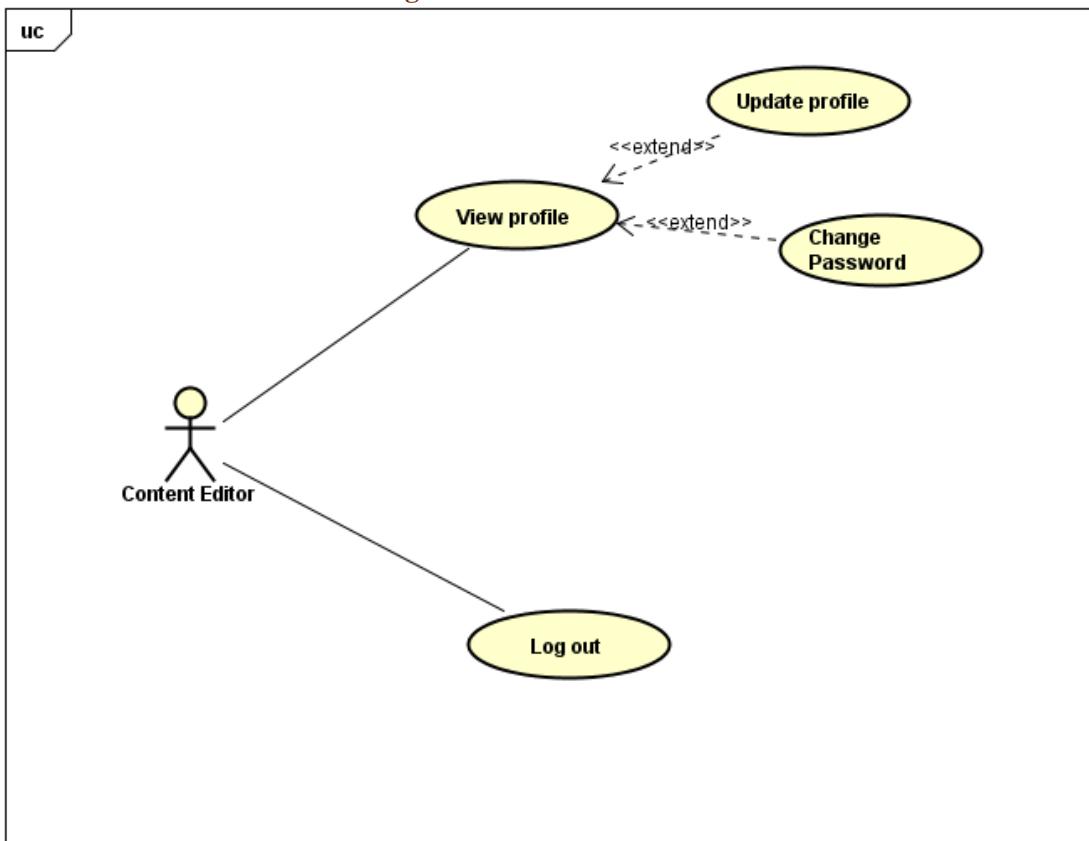


Figure 4-11 Private Account Management

No.	Use-case name	Actor	Description
1	View Profile	Content Editor	Viewing detail information of their profile
2	Update Profile	Content Editor	Updating user's profile
3	Change Password	Content Editor	Changing user's password
4	Log Out	Content Editor	Logging out of system

4.2.7.3.2 Lesson Management

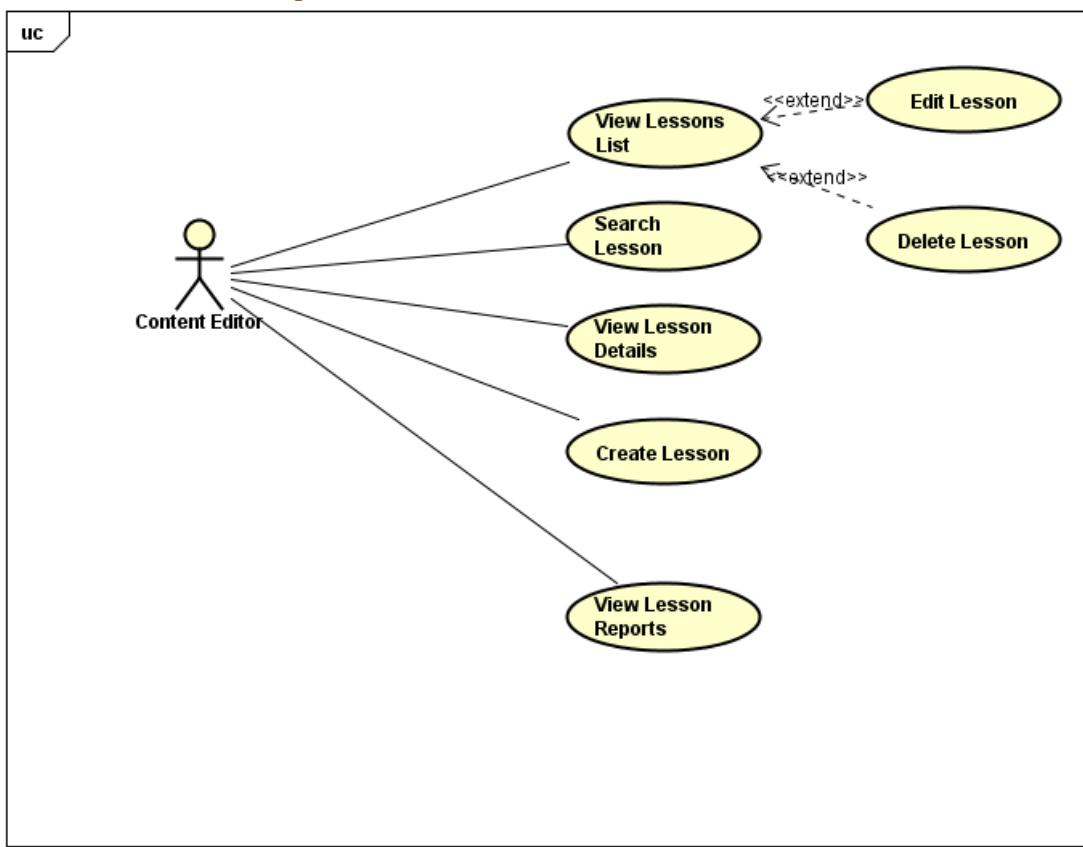


Figure 4-12 Lesson Management

No.	Use-case name	Actor	Description
1	View Lessons List	Content Editor	Viewing lesson list of courses
2	Search Lesson	Content Editor	Searching lesson
3	View Lesson Details	Content Editor	View detail of lesson
4	Edit Lesson	Content Editor	Editing content of lesson
5	Delete Lesson	Content Editor	Deleting lesson
6	Create Lesson	Content Editor	Creating new lesson of course
7	View Lesson Reports	Content Editor	Viewing list report about wrong content from member

4.2.7.3.3 Question Management

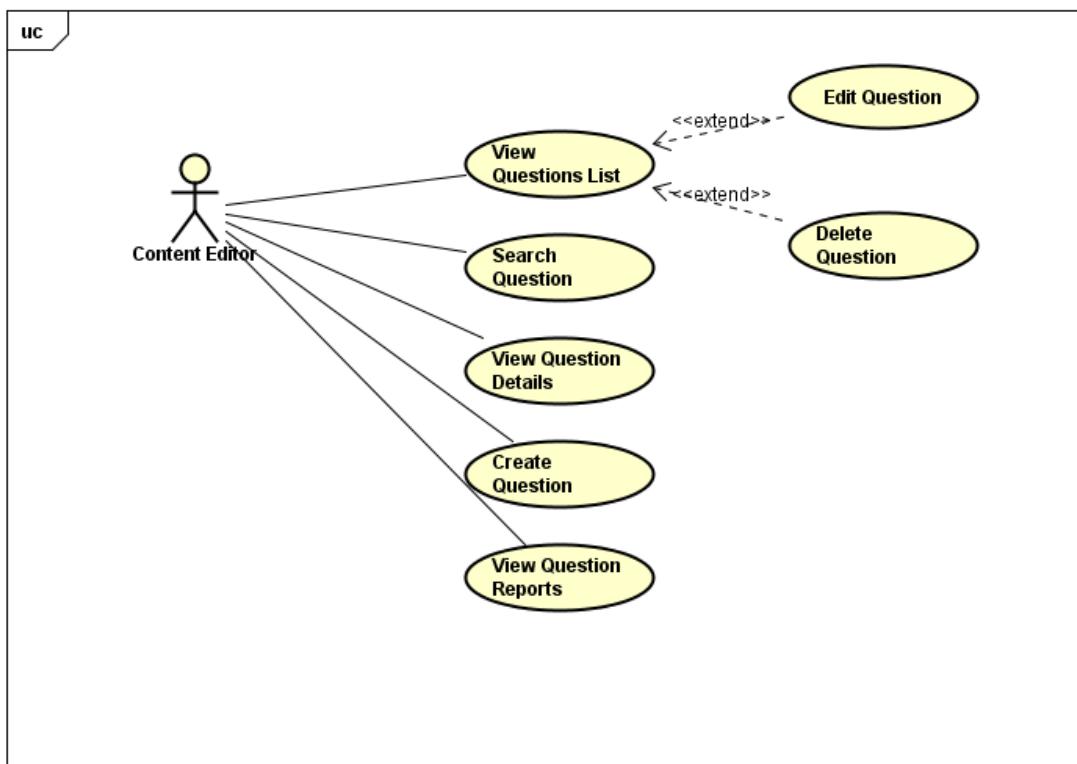


Figure 4-13 Question Management

No.	Use-case name	Actor	Description
1	View Questions List	Content Editor	Viewing question list
2	Search Question	Content Editor	Searching question
3	View Question Details	Content Editor	View detail of question
4	Edit Question	Content Editor	Editing content of question
5	Delete Question	Content Editor	Deleting question
6	Create Question	Content Editor	Creating new question of course
7	View Question Report	Content Editor	Viewing list report about wrong content from member

4.2.8 Logical View

4.2.8.1 Overview

Logical View includes Package diagram and Class diagram. Package diagram describes the organization of packages and elements. Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system

- Controller contain the interface between
 - ✓ Associated models
 - ✓ Associated views
 - ✓ The input devices (e.g., keyboard, pointing device, time).
 - ✓ Send commands to the model to update the model's state.
- Model is:
 - ✓ the domain-specific software simulation
 - ✓ Or implementation of the application's central structure.
- View deal with everything graphical
 - ✓ Requests data from their model
 - ✓ Display the data

4.2.8.2 Architecturally Significant Design Packages

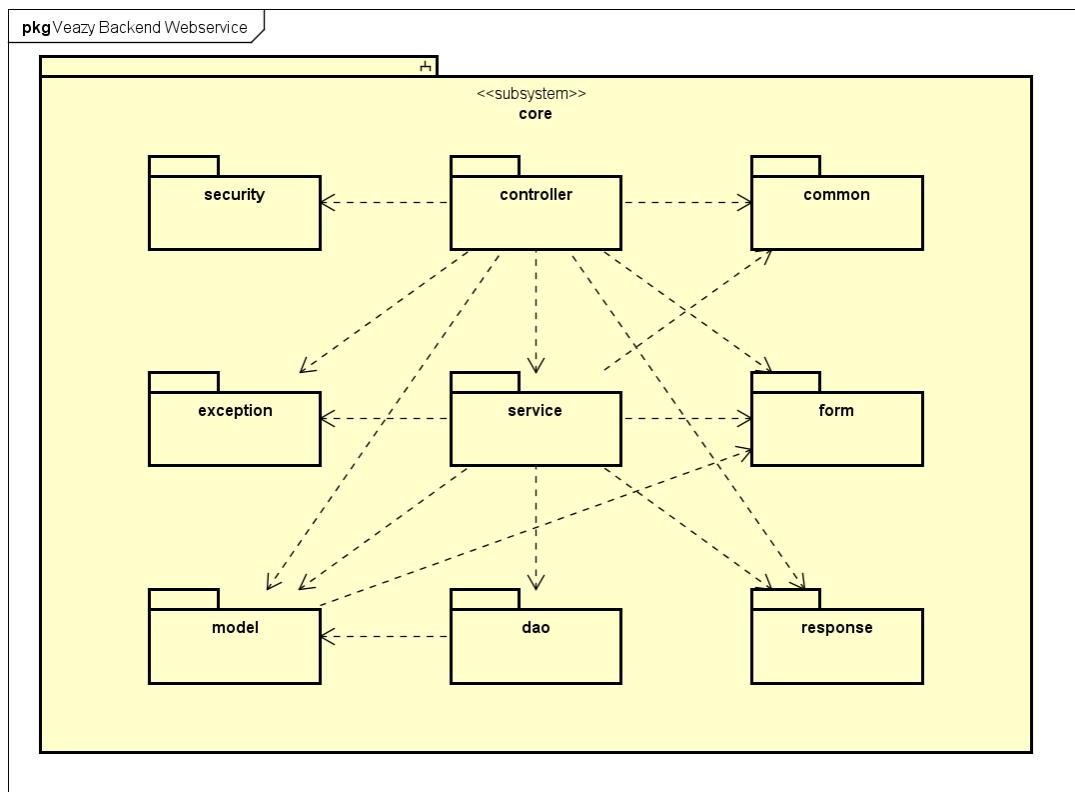


Figure 4-17: Package Diagram

- Controller

No.	Controller class	Role
1	CoreController	Processes common core api requests
2	CourseController	Processes api requests related to course
3	TestController	Processes api requests related to test
4	LessonController	Processes api requests related to lesson
5	QuestionController	Processes api requests related to question
6	ReportController	Processes api requests related to report
7	StatsController	Processes api requests related to statistic
8	UserController	Processes api requests related to user

- Service

No.	Service class	Role
1	CourseServiceImpl	Contains business logical functions for processing course data
2	TestServiceImpl	Contains business logical functions for processing test data
3	LessonServiceImpl	Contains business logical functions for processing lesson data
4	QuestionBankServiceImpl	Contains business logical functions for processing question bank data
5	QuestionServiceImpl	Contains business logical functions for processing question data
6	ReportServiceImpl	Contains business logical functions for processing report data
7	StatsServiceImpl	Contains business logical functions for processing statistic data
8	UserServiceImpl	Contains business logical functions for processing user data

- Model

No.	Model class	Role
1	AnswerModel	Stores data of a single answer of a question
2	BasicModel	Contains common fields of model
3	CourseModel	Stores data of a course (as known as a level)
4	TestAnswerModel	Stores data of a single answer of a question in a generated test
5	TestModel	Stores data of a test
6	TestQuestionModel	Stores data of a question in a generated test
7	LessonModel	Stores data of a lesson
8	LessonVersionModel	Stores data of a version of lesson
9	QuestionModel	Stores data of a question in the question bank

10	ReportModel	Stores data of a report
11	UserModel	Stores data of an user

- **DAO**

No.	DAO class	Role
1	HibernateCourseDao	Provides methods to manage course data in database
2	HibernateTestDao	Provides methods to manage test data in database
3	HibernateLessonDao	Provides methods to manage lesson data in database
4	HibernateLessonVersion Dao	Provides methods to manage versions of lesson in database
5	HibernateQuestionDao	Provides methods to manage question data in database
6	HibernateReportDao	Provides methods to manage report data in database
7	HibernateUserDao	Provides methods to manage user data in database
8	HibernateAnswerDao	Provides methods to manage answers data in a question in database

- **Exception**

No.	Exception class	Role
1	CorruptedFormException	Exception throwed when user submitted a corrupted form i.e. form that does not have enough expected data
2	EmailExpectedException	Exception throwed when submitted form does not contain email
3	InvalidEmailException	Exception throwed when submitted form contains invalid email
4	PasswordExpectedException	Exception throwed when submitted form does not contain password

5	PasswordIncorrectException	Exception throwed when submitted form contains incorrect username or password
6	UsernameExpectedException	Exception throwed when submitted form does not contain username
7	NullOrEmptyContentException	Exception throwed when submitted data is null or empty

- Form

No.	Form class	Role
1	AnswerForm	Answer form
2	ChangeRoleForm	Form for requesting change user role
3	ChgpwdForm	Form for requesting change password
4	CreateTestSinglePartForm	Form for generating a single-skill test
5	CreateLessonForm	Form for creating a test
6	TestPartForm	Form for generating a test
7	FileUploadForm	Upload file form
8	LoginForm	Login form
9	QuestionForm	Question form
10	RegisterForm	Register form
11	ReportForm	Report form
12	SubmitTestForm	Test form for submission
13	SubmitQuestionForm	Question form for test submission
14	SubmitAnswerForm	Answer of a question form for test submission
15	UpdateLessonForm	Lesson information form for updating a lesson
16	UpdateUserForm	User information form for updating an user

- Response

No.	Response class	Role
1	AddQuestionResponse	Response for adding a new question api
2	AnswerResponse	Response for an answer in a question
3	BriefAnswerResponse	Response for a brief answer in a brief question in a generated test
4	BriefLessonResponse	Response for a brief lesson
5	BriefQuestionResponse	Response for a brief question in a generated test
6	CreateLessonResponse	Response for creating a lesson api
7	TestAnswerResponse	Response for an answer of a question in a generated test
8	TestPartResponse	Response for a generated test
9	TestQuestionResponse	Response for a question in a generated test
10	TestResultResponse	Response for test result
11	TestSinglePartResponse	Response for generating a single-skill test api
12	GetCourseResponse	Response for getting list courses api
13	GetTestResponse	Response for getting a taken test api
14	GetLearnerTestsResponse	Response for getting learner tests api
15	GetLessonResponse	Response for getting a lesson api
16	GetLessonVersionResponse	Response for getting a lesson version api
17	GetListUsersResponse	Response for getting list users api
18	GetUserResponse	Response for getting a user api
19	LoginResponse	Response for login api
20	QuestionResponse	Response for a question
21	Response	Response form for all api
22	ResponseCode	Return code for each api
23	StatsCourseAvgResponse	Response for getting average result of tests via course statistic api
24	StatsLastTestResponse	Response for getting last test statistic api

25	StatsLessonsResponse	Response for getting lesson statistic api
26	StatsQuestionsResponse	Response for getting questions statistic api
27	StatsSkillAvgResponse	Response for getting average result of tests via skill statistic api
28	StatsUsersResponse	Response for getting users statistic api
29	UploadFileResponse	Response for uploading api

- **Security**

No.	Security class	Role
1	Veazy403Handler	Handles 403 forbidden error
2	VeazyAuthenEntryPoint	Handles anonymous requests
3	VeazyAuthenFailureHandler	Handles authentication failure requests
4	VeazyAuthenFilter	Filters authentication requests
5	VeazyAuthenProvider	Provides authentication mechanism
6	VeazyAuthenSuccessHandler	Handles authentication success requests

- **Common**

No.	Common class	Role
1	Const	Contains common system constants
2	Utils	Contains common utility functions
3	JsonUtils	Contains json utility functions
4	HttpUtils	Contains http utility functions
5	HtmlUtils	Contains html utility functions

4.2.9 Process View

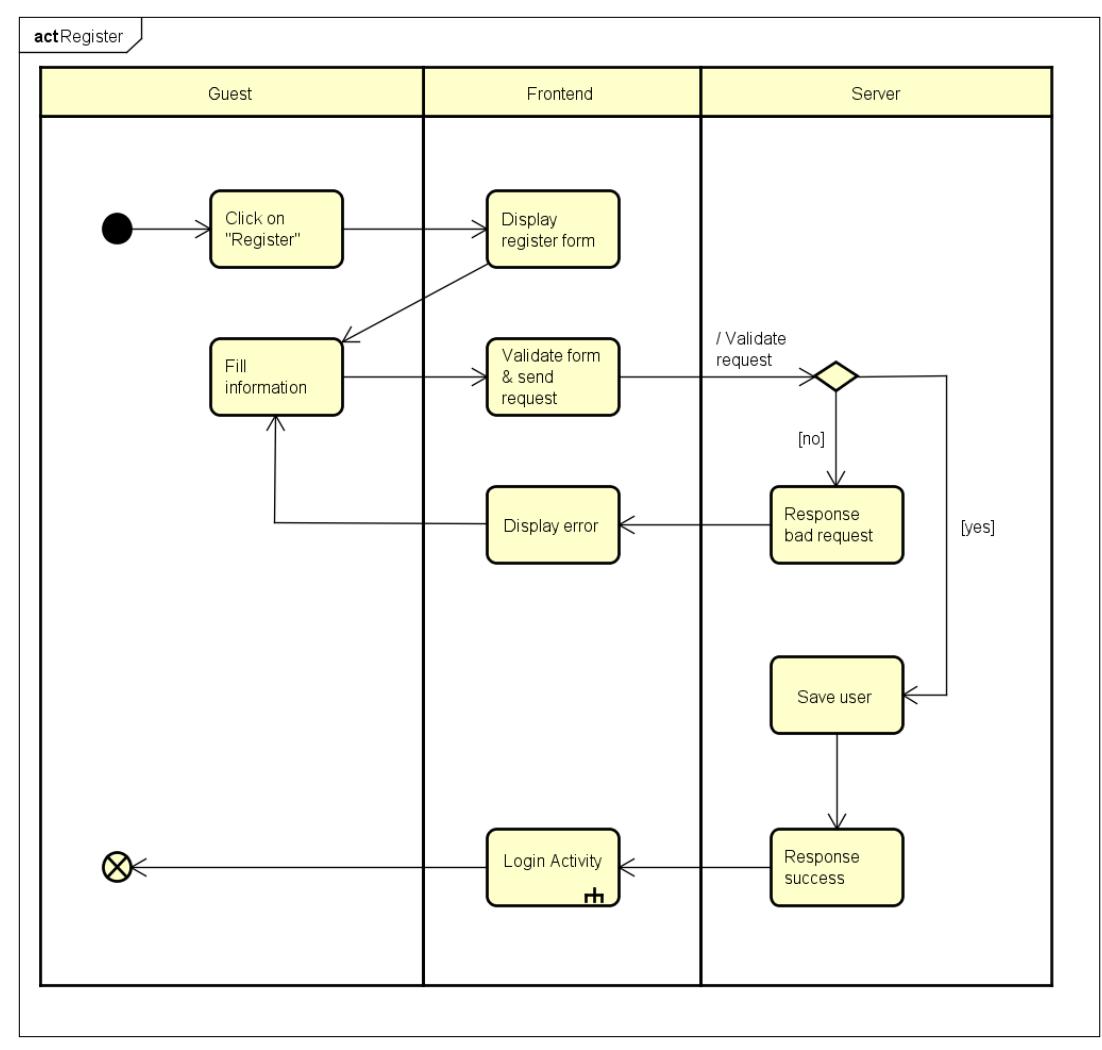


Figure 4-14 Register activity

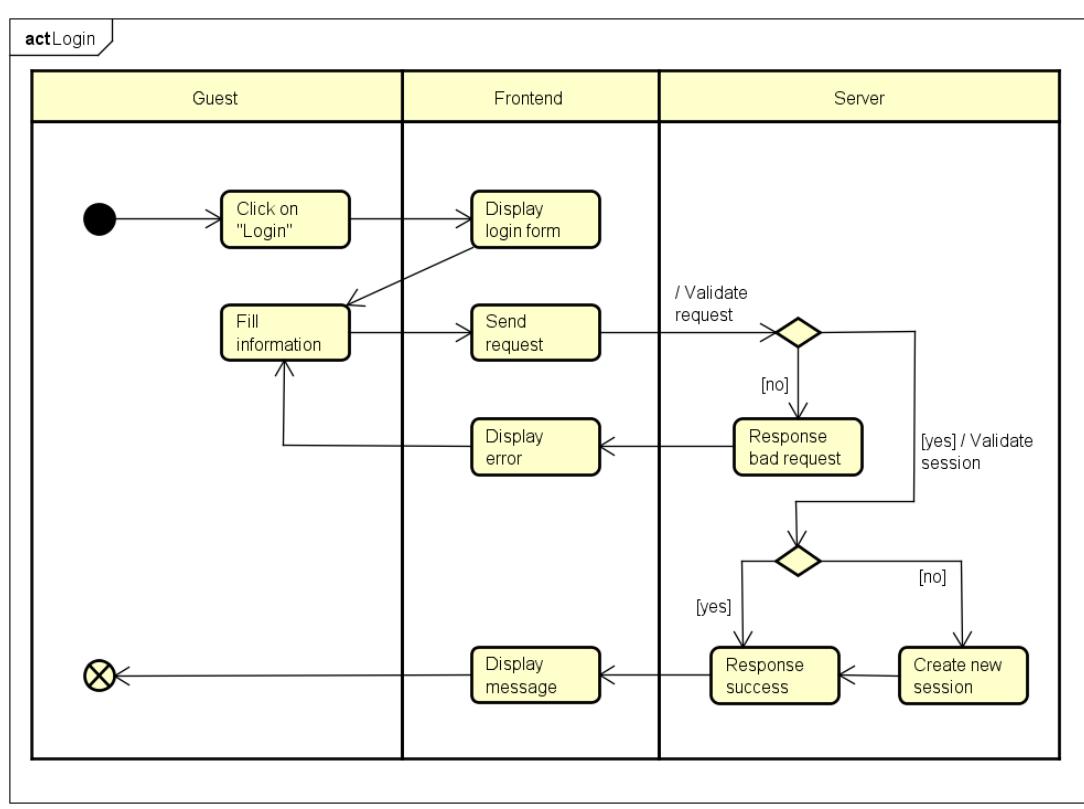


Figure 4-15 Login activity

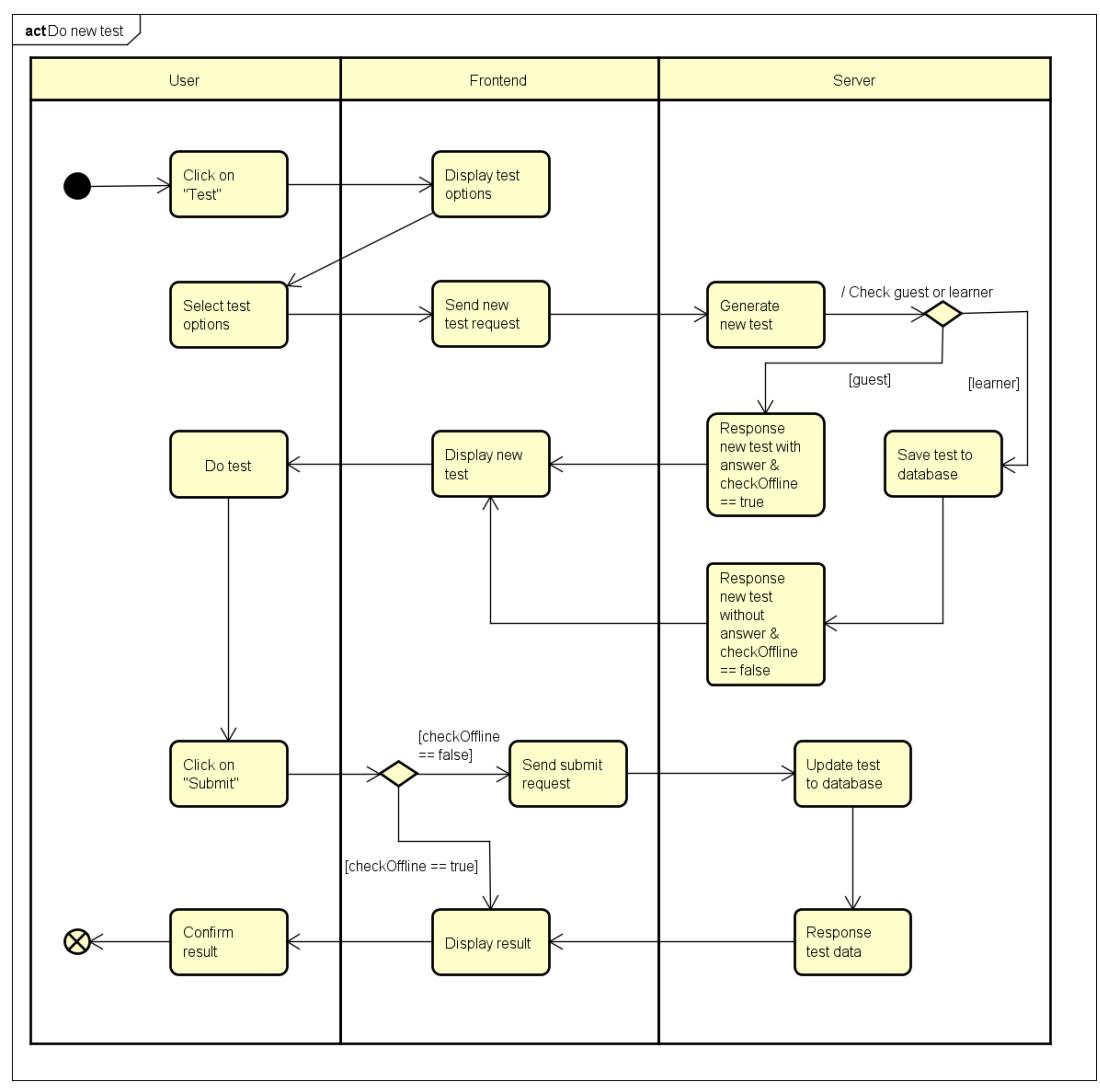


Figure 4-16 Do test activity

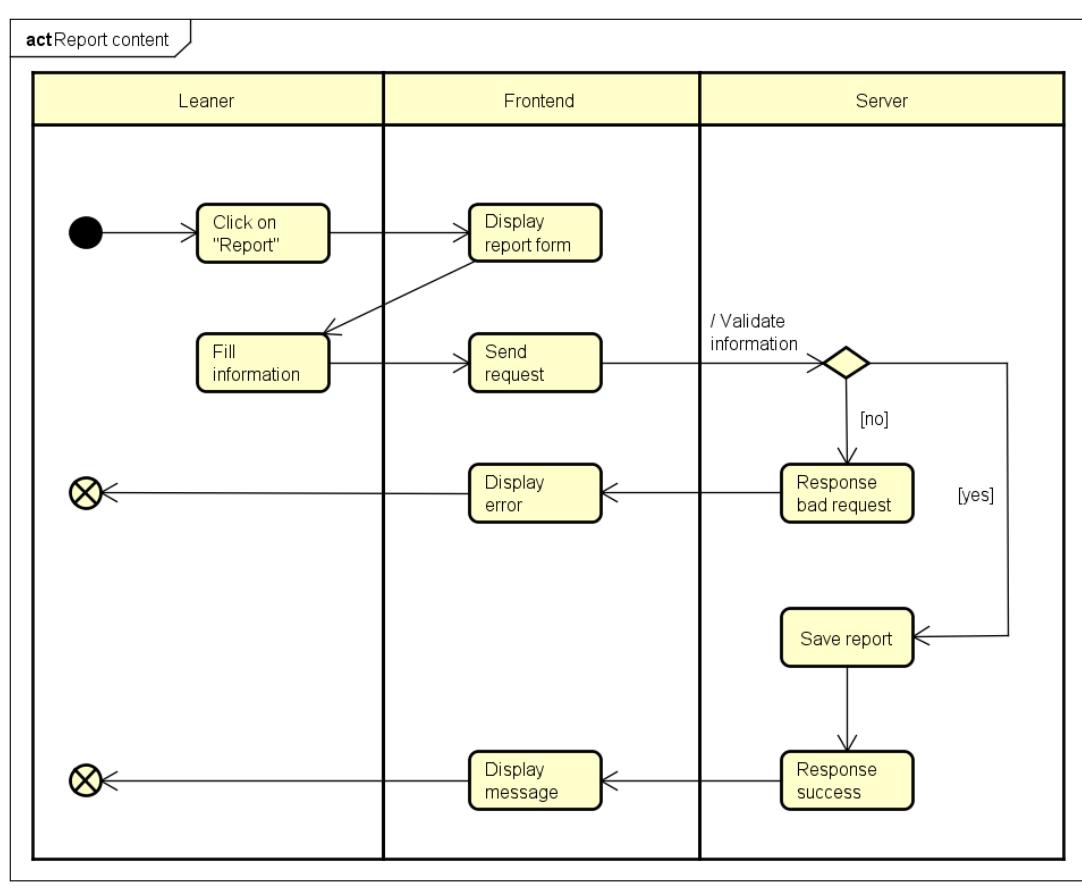


Figure 4-17 Report content activity

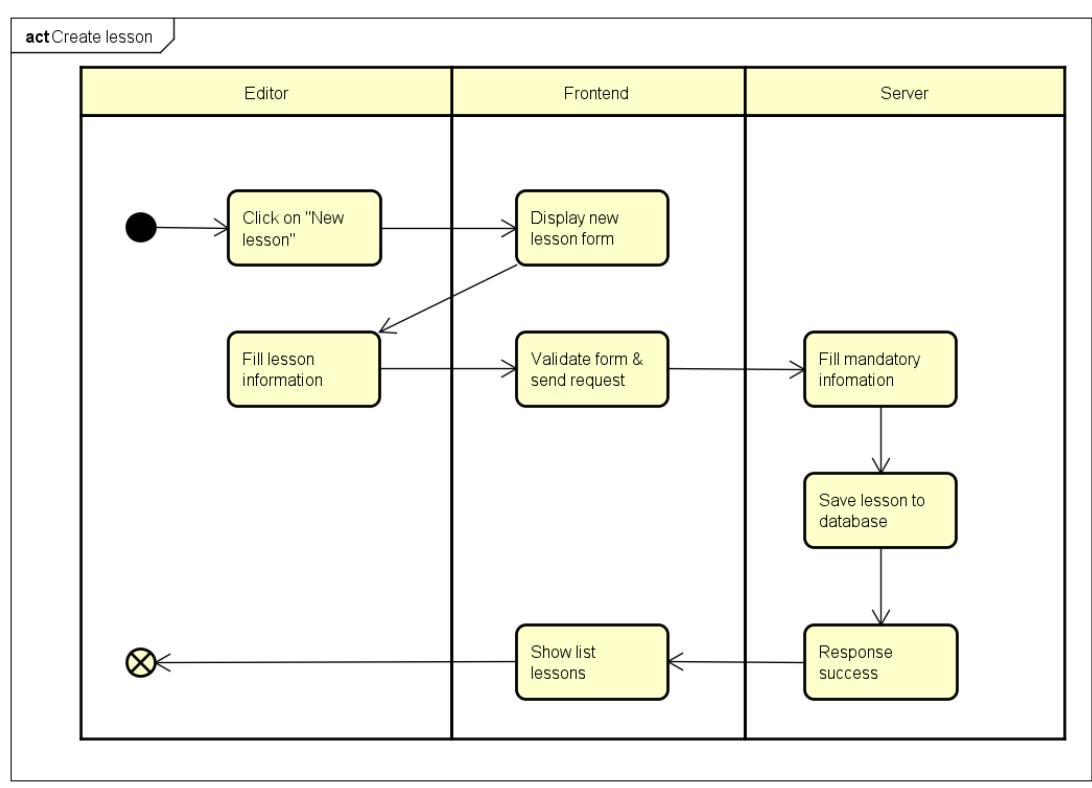


Figure 4-18 Create lesson activity

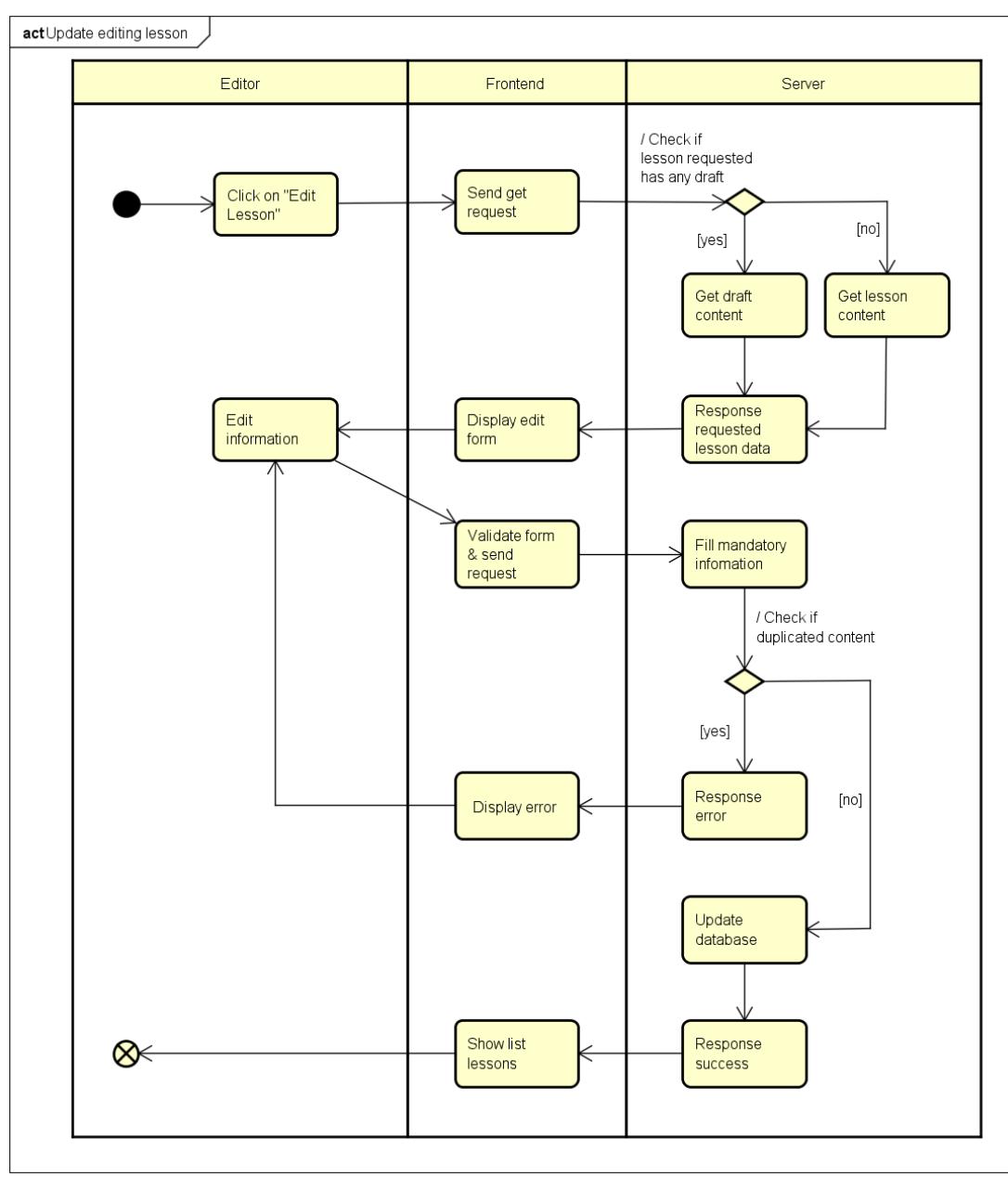


Figure 4-19 Update lesson activity

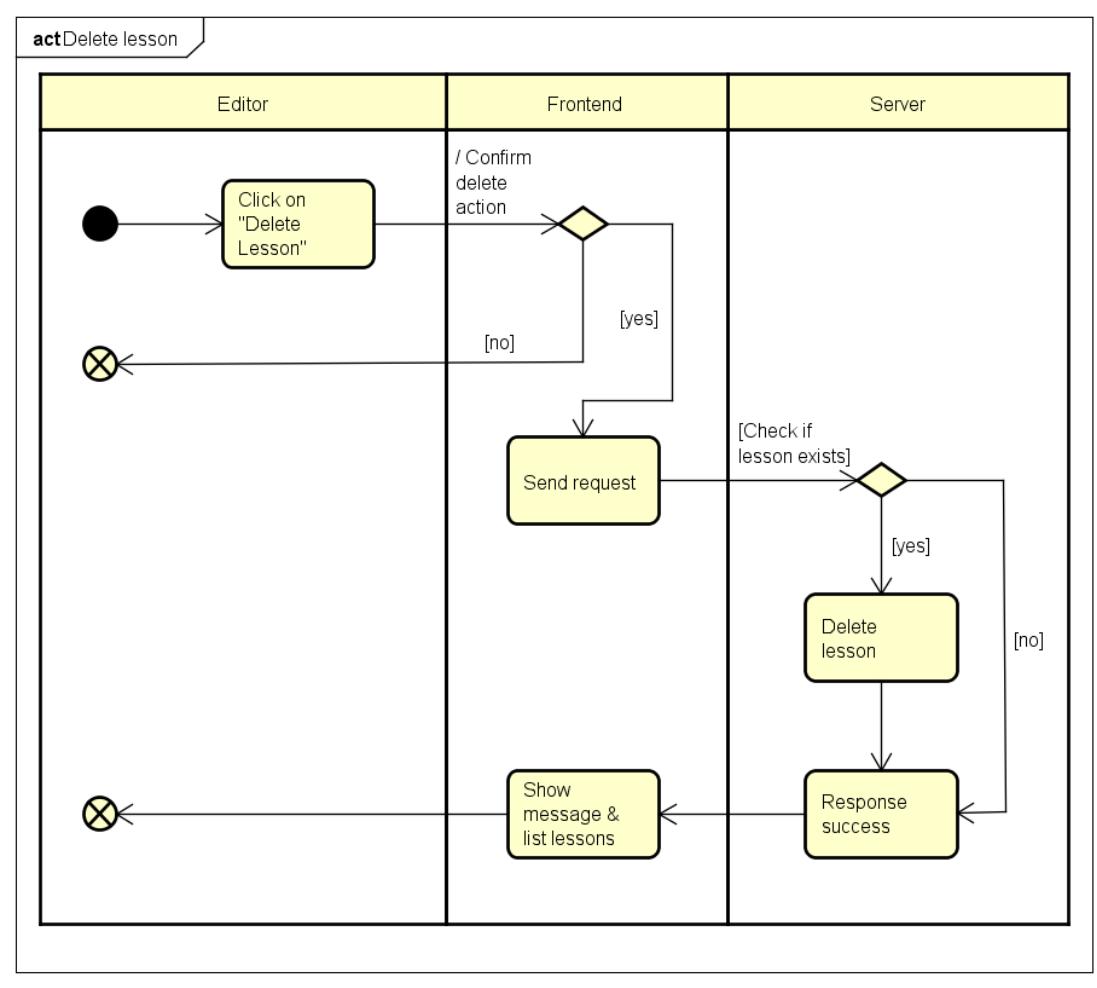


Figure 4-20 Delete lesson activity

4.2.10 Deployment view

Deployment view of Veazy system

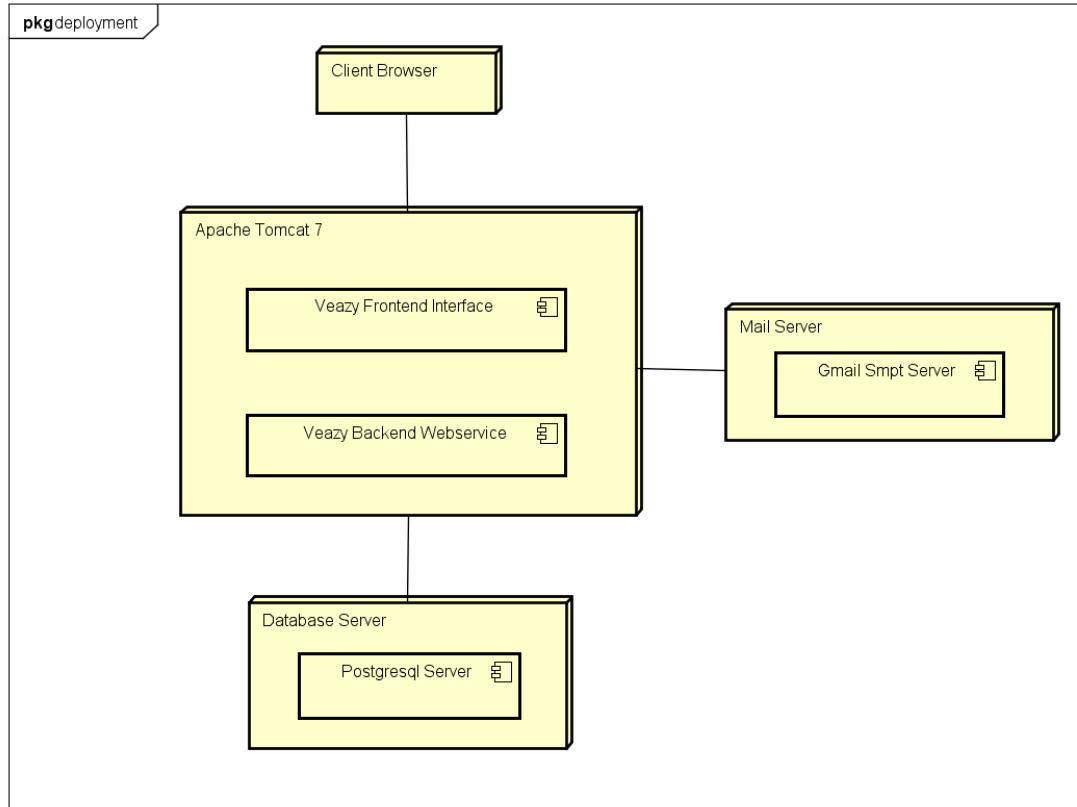


Figure 4-21 Deployment diagram

No.	Name	Description
1	Postgresql Server	Postgresql server to persist data. Postgresql 9.4 or higher
2	Mail Server	Mail server to send email to user
3	Web Server	Web server hosting system
4	Client Browser	Web browser to display system. Firefox 20, Chrome 36 or higher.

4.2.11 Class Design

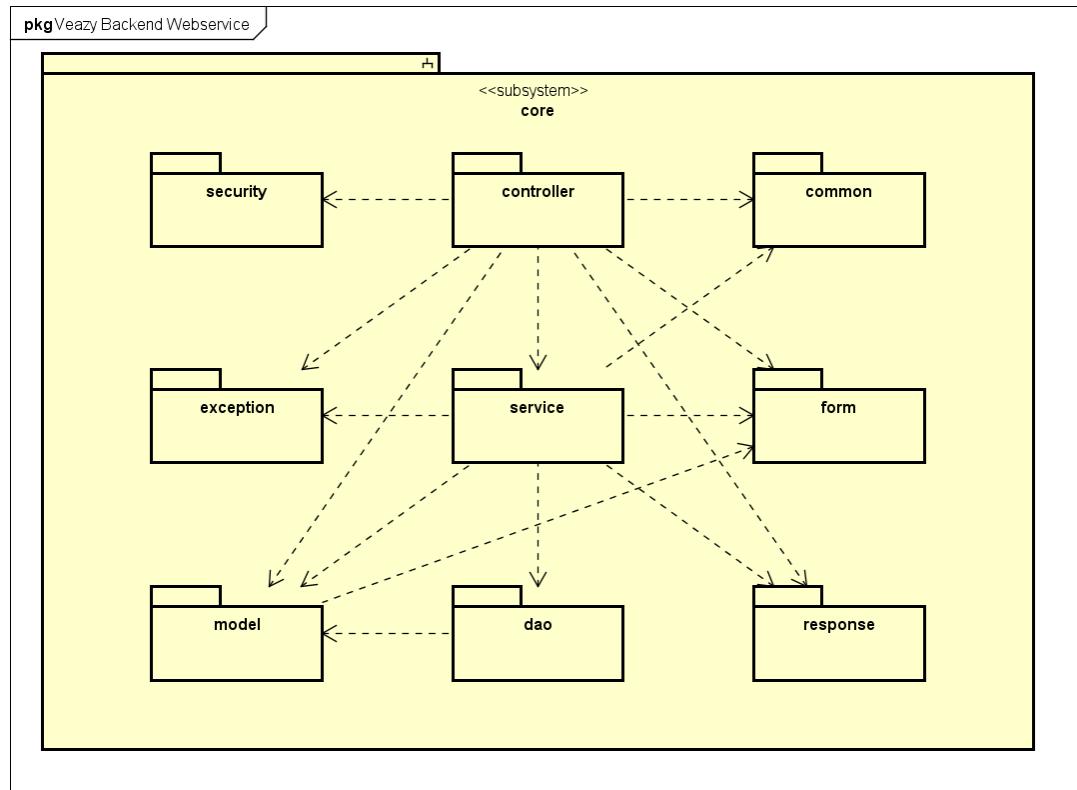


Figure 0: List Packages

No	Package	Language	Description
1	core	JAVA	Contains classes related to core functions
2	common	JAVA	Contains common shared classes
3	controller	JAVA	Contains Spring controllers
4	dao	JAVA	Contains Hibernate data access object classes
5	exception	JAVA	Contains user-defined exceptions
6	form	JAVA	Contains pojo data forms
7	model	JAVA	Contains Spring entity classes
8	response	JAVA	Contains data response classes
9	security	JAVA	Contain classes related to security
10	service	JAVA	Contains Spring service classes

4.2.11.1 core.common Package

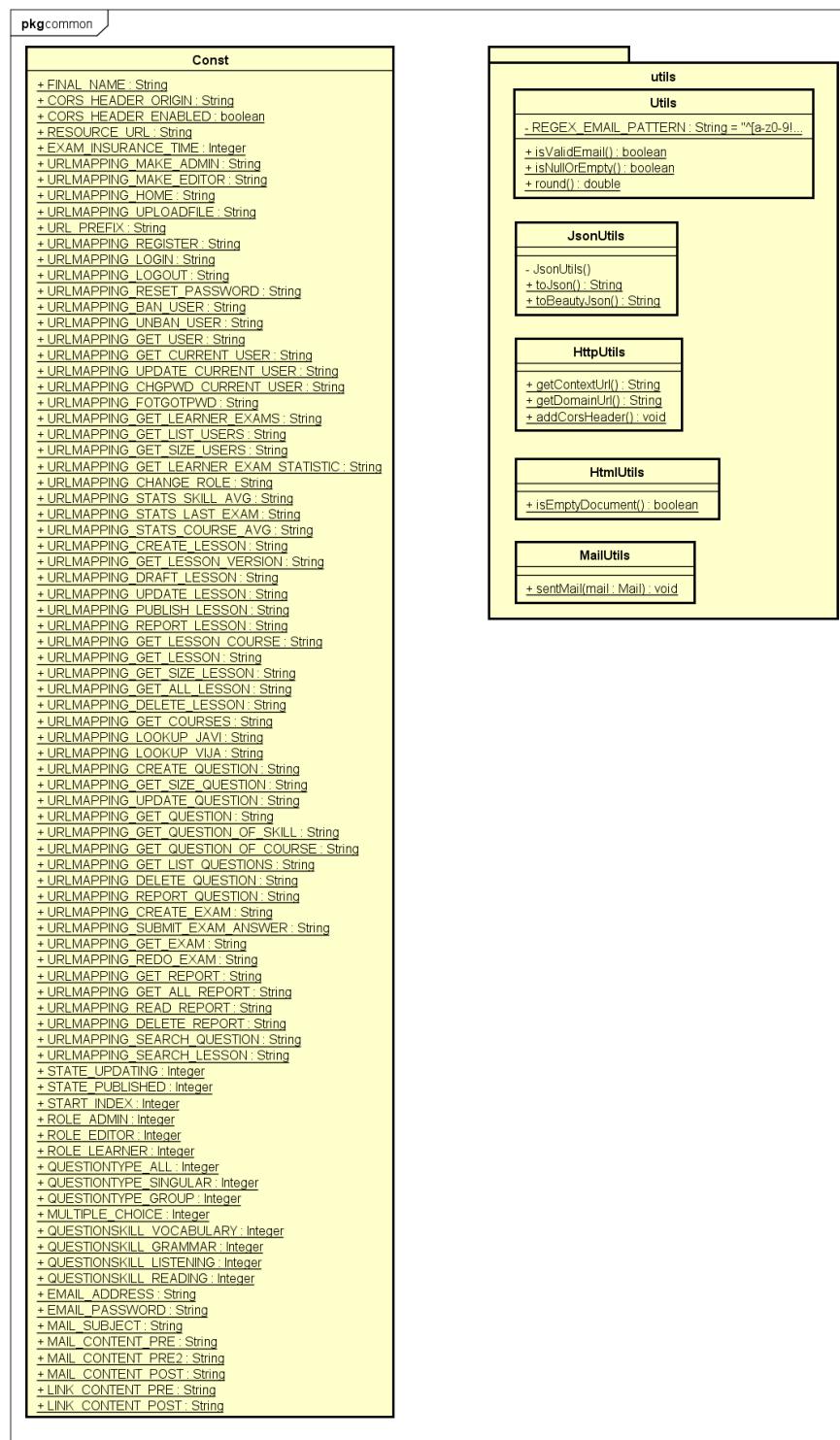


Figure 1: Class diagram core.common package

No	Class Name	Description
1	Const	Contain constant variable
2	Utils	Contain common utility function
3	JsonUtils	Contain utility function about JSON
4	HttpUtils	Contain utility function about http
5	HtmlUtils	Contain utility function about html
6	MailUtils	Contain utility function about mail

4.2.11.1.1 Const class**Attributes**

No	Attribute	Type	Value
1	FINAL_NAME	String	Veazy
2	CORS_HEADER_ORIGIN	String	Origin
3	CORS_HEADER_ENABLED	boolean	TRUE
4	RESOURCE_URL	String	/res
5	EXAM_INSURANCE_TIME	Integer	30
6	URLMAPPING_MAKE_ADMIN	String	/mkad/{user_id}
7	URLMAPPING_MAKE_EDITOR	String	/mked/{user_id}
8	URLMAPPING_HOME	String	/
9	URLMAPPING_UPLOADFILE	String	/uploadfile
10	URL_PREFIX	String	/api
11	URLMAPPING_REGISTER	String	URL_PREFIX + "/register"
12	URLMAPPING_LOGIN	String	URL_PREFIX + "/login"
13	URLMAPPING_LOGOUT	String	URL_PREFIX + "/logout"
14	URLMAPPING_RESET_PASSWORD	String	URL_PREFIX + "/users/reset_password"
15	URLMAPPING_BAN_USER	String	URL_PREFIX + "/users/ban/{user_id}"

16	URLMAPPING_UNBAN_USER	String	URL_PREFIX + "/users/unban/{user_id}"
17	URLMAPPING_GET_USER	String	URL_PREFIX + "/users/{user_id}"
18	URLMAPPING_GET_CURRENT_USER	String	URL_PREFIX + "/user"
19	URLMAPPING_UPDATE_CURRENT_USER	String	URL_PREFIX + "/user/update"
20	URLMAPPING_CHGPWD_CURRENT_USER	String	URL_PREFIX + "/user/chgpwd"
21	URLMAPPING_FOTGOTPWD	String	URL_PREFIX + "/user/forgotpwd"
22	URLMAPPING_GET_LEARNER_EXAMS	String	URL_PREFIX + "/user/exams"
23	URLMAPPING_GET_LIST_USERS	String	URL_PREFIX + "/users"
24	URLMAPPING_GET_SIZE_USERS	String	URL_PREFIX + "/users/count"
25	URLMAPPING_GET_LEARNER_EXAM_STATISTIC	String	URL_PREFIX + "/users/{user_id}/exams/statistics"
26	URLMAPPING_CHANGE_ROLE	String	URL_PREFIX + "/users/change_roll/{user_id}"
27	URLMAPPING_STATS_SKILL_AVG	String	URL_PREFIX + "/stats/skill/avg"
28	URLMAPPING_STATS_LAST_EXAM	String	URL_PREFIX + "/stats/exam/last/{number}"
29	URLMAPPING_STATS.Course_AVG	String	URL_PREFIX + "/stats/course/avg"
30	URLMAPPING_CREATE_LESSON	String	URL_PREFIX + "/lessons/new"
31	URLMAPPING_GET_LESSON_VERSION	String	URL_PREFIX + "/lessons/{lesson_id}/ver{version}"
32	URLMAPPING_DRAFT_LESSON	String	URL_PREFIX + "/lessons/savedraft/{lesson_id}"
33	URLMAPPING_UPDATE_LESSON	String	URL_PREFIX + "/lessons/update/{lesson_id}"
34	URLMAPPING_PUBLISH_LESSON	String	URL_PREFIX + "/lessons/publish/{lesson_id}"

35	URLMAPPING_REPORT_LESSON	String	URL_PREFIX + "/lessons/report/{lesson_id}"
36	URLMAPPING_GET_LESSON_COURSE	String	URL_PREFIX + "/courses/{course_id}/lessons"
37	URLMAPPING_GET_LESSON	String	URL_PREFIX + "/lessons/{lesson_id}"
38	URLMAPPING_GET_SIZE_LESSON	String	URL_PREFIX + "/lessons/count"
39	URLMAPPING_GET_ALL_LESSON	String	URL_PREFIX + "/lessons"
40	URLMAPPING_DELETE_LESSON	String	URL_PREFIX + "/lessons/delete/{lesson_id}"
41	URLMAPPING_GET_COURSES	String	URL_PREFIX + "/courses"
42	URLMAPPING_LOOKUP_JAVI	String	URL_PREFIX + "/dict/javi/"
43	URLMAPPING_LOOKUP_VIJA	String	URL_PREFIX + "/dict/vija/"
44	URLMAPPING_CREATE_QUESTION	String	URL_PREFIX + "/questions/new"
45	URLMAPPING_GET_SIZE_QUESTION	String	URL_PREFIX + "/questions/count"
46	URLMAPPING_UPDATE_QUESTION	String	URL_PREFIX + "/questions/update/{question_id}"
47	URLMAPPING_GET_QUESTION	String	URL_PREFIX + "/questions/{question_id}"
48	URLMAPPING_GET_QUESTION_OF_SKILL	String	URL_PREFIX + "/skill/{skill_id}/questions"
49	URLMAPPING_GET_QUESTION_OF.Course	String	URL_PREFIX + "/courses/{course_id}/questions"
50	URLMAPPING_GET_LIST_QUESTIONS	String	URL_PREFIX + "/questions"
51	URLMAPPING_DELETE_QUESTION	String	URL_PREFIX + "/questions/delete/{question_id}"
52	URLMAPPING_REPORT_QUESTION	String	URL_PREFIX + "/questions/report/{question_id}"
53	URLMAPPING_CREATE_EXAM	String	URL_PREFIX + "/exams"
54	URLMAPPING_SUBMIT_EXAM_ANSWER	String	URL_PREFIX + "/exams/submit"

55	URLMAPPING_GET_EXAM	String	URL_PREFIX + "/exams/{exam_id}"
56	URLMAPPING_REDO_EXAM	String	URL_PREFIX + "/exams/{exam_id}/redo"
57	URLMAPPING_GET_REPORT	String	URL_PREFIX + "/report/get/{report_id}"
58	URLMAPPING_GET_ALL_REPORT	String	URL_PREFIX + "/report/all"
59	URLMAPPING_READ_REPORT	String	URL_PREFIX + "/report/read/{report_id}"
60	URLMAPPING_DELETE_REPORT	String	URL_PREFIX + "/report/delete/{report_id}"
61	URLMAPPING_SEARCH_QUESTION	String	URL_PREFIX + "/search/question"
62	URLMAPPING_SEARCH_LESSON	String	URL_PREFIX + "/search/lesson"
63	UPDATING	Integer	1
64	PUBLISHED	Integer	2
65	START_INDEX	Integer	1
66	ROLE_ADMIN	Integer	1
67	ROLE_EDITOR	Integer	2
68	ROLE_LEARNER	Integer	3
69	QUESTIONTYPE_ALL	Integer	0
70	QUESTIONTYPE_SINGULAR	Integer	1
71	QUESTIONTYPE_GROUP	Integer	2
72	MULTIPLE_CHOICE	Integer	1
73	QUESTIONSKILL_VOCABULARY	Integer	1
74	QUESTIONSKILL_GRAMMAR	Integer	2
75	QUESTIONSKILL_LISTENING	Integer	3
76	QUESTIONSKILL_READING	Integer	4
77	EMAIL_ADDRESS	String	veazysystem@gmail.com

78	EMAIL_PASSWORD	String	veazy123
79	MAIL SUBJECT	String	Veazy Reset Password
80	MAIL_CONTENT_PRE	String	Hey
81	MAIL_CONTENT_PRE2	String	,\nSomeone has requested a new password for your Veazy account.\n
82	MAIL_CONTENT_POST	String	If you didn't make this request then you can safely ignore this email.\nBest,\n\nThe Veazy Team
83	LINK_CONTENT_PRE	String	<a href=\"
84	LINK_CONTENT_PRE	String	\">Reset Password

4.2.11.1.2 Utils class

Attributes

No	Attribute	Type	Note	Description
1	REGEX_EMAIL_PATTERN	String	private	Regex pattern to determine an email

Methods

No	Method	Description
1	isValidEmail()	Check if a string is a valid email or not.
2	isNullOrEmpty()	Check if a string is null or empty.
3	round()	round a decimal number

4.2.11.1.3 JsonUtils class

Methods

No	Method	Description
1	toJson()	Convert an object to a json format
2	toBeautyJson()	Convert an object to a beautiful json format

4.2.11.1.4 HttpUtils Class

Methods

No	Method	Description
1	getContextUrl()	Extracts context url from requested url
2	addCorsHeader()	Adds CORS headers if enabled

4.2.11.1.5 HtmlUtils Class

Methods

No	Method	Description
1	isEmptyDocument()	Check if Html document is empty

4.2.11.1.6 MailUtils Class

Methods

No	Method	Description
1	sentMail()	Send an email

4.2.11.2 core.controller Package

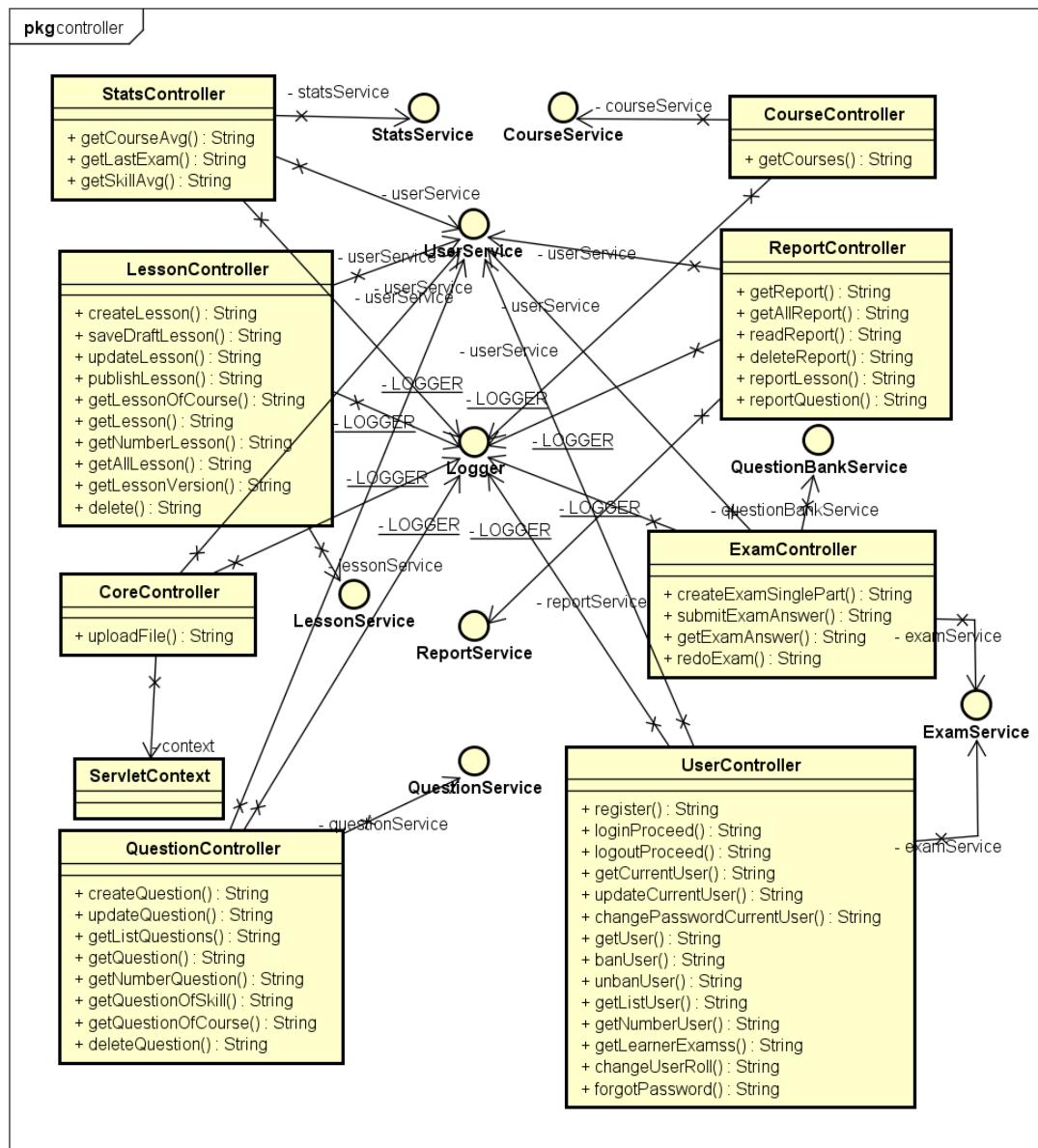


Figure 2: Class diagram `core.controller` package

No	Class Name	Description
1	CoreController	Control core API
2	CourseController	Control course API
3	ExamController	Control exam API
4	LessonController	Control lesson API

5	QuestionController	Control question API
6	ReportController	Control report API
8	StatsController	Control user statistic API
9	UserController	Control user API

4.2.11.2.1 CoreController Class

Methods

No	Method	Description
2	uploadFile()	To upload file

4.2.11.2.2 CourseController Class

Methods

No	Method	Description
1	getCourses()	Return all courses

4.2.11.2.3 ExamController Class

Methods

No	Method	Description
1	createExamSinglePart()	Generate an exam for learner and return the exam.
2	submitExamAnswer()	Submit exam answers.
3	getExamAnswer()	Return the answer of an exam
4	redoExam()	Return an exam the user had done.

4.2.11.2.4 LessonController Class

Methods

No	Method	Description
1	createLesson	Create a lesson.
2	saveDraftLesson	Save lesson as draft.
3	updateLesson	Update a lesson.
4	publishLesson	Make updating lesson version to publish state.
5	getLessonOfCourse	Return all lesson of a course.
6	getLesson	Return a lesson.

7	getNumberLesson	Return the number of lesson in system.
8	getAllLesson	Return all lesson of system
9	getLessonVersion	Return the version of lesson
10	delete	Delete a lesson

4.2.11.2.5 QuestionController Class

Methods

No	Method	Description
1	createQuestion()	Create a question.
2	updateQuestion()	Update a question.
3	getListQuestions()	Return list of question.
4	getQuestion()	Return a question.
5	getNumberQuestion()	Return the number of question in system.
6	getQuestionOfSkill()	Return all question by a skill.
7	getQuestionOfCourse()	Return all question in a course.
8	deleteQuestion()	Delete a question.

4.2.11.2.6 ReportController Class

Methods

No	Method	Description
1	getReport ()	Return a report
2	getAllReport ()	Return all report
3	readReport ()	Make a report as read state
4	deleteReport ()	Delete a report
5	reportLesson	Report a lesson.
6	reportQuestion()	Report a question.

4.2.11.2.7 StatsController Class Methods

No	Method	Description
1	getCourseAvg	Return the average point of a course.
2	getLastExam	Return the last exam of user
3	getSkillAvg	Return the average point of a test skill.

4.2.11.2.8 UserController Class Methods

No	Method	Description
1	register()	Sign up an account
2	loginProceed()	Login to system
3	logoutProceed()	Logout system
4	getCurrentUser()	Return current user information
5	updateCurrentUser()	Update user information
6	changePasswordCurrentUser()	Change password
7	getUser()	Return an user information
8	banUser()	Ban a user.
9	unbanUser()	Unban a user.
10	getListUser()	Return list of the user.
11	getNumberUser()	Return the number of user in system.
12	getLearnerExamss()	Return the done exam of a learner.
13	changeUserRoll()	Change user role.

4.2.11.3 core.dao Package

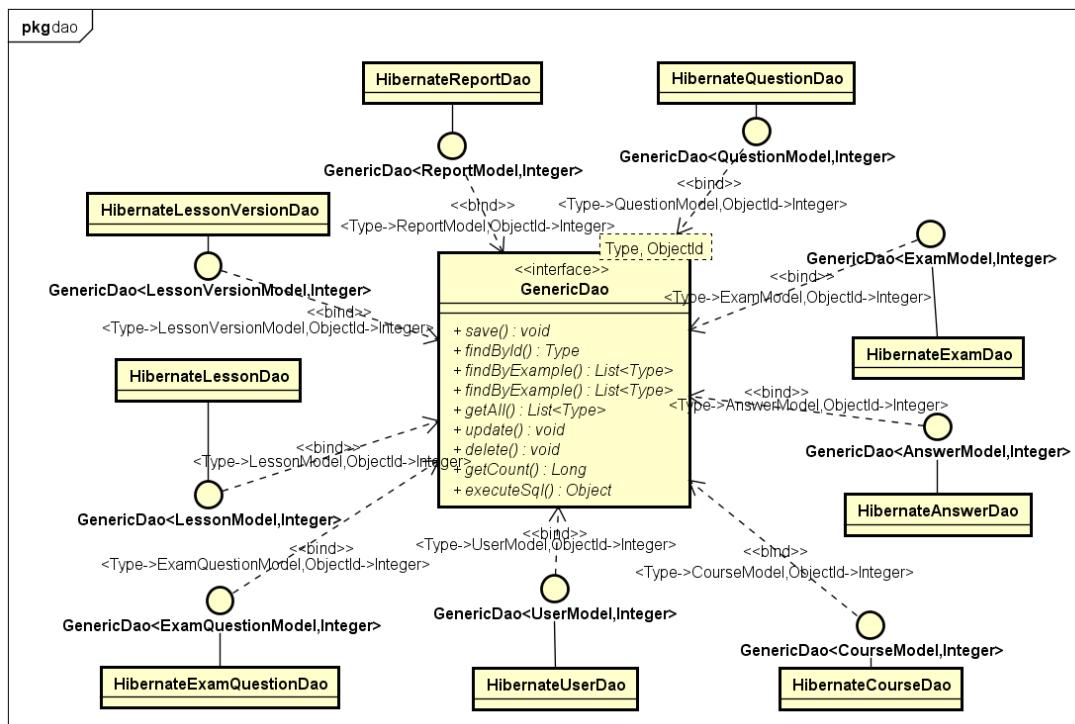


Figure 3: Class diagram `core.dao` package

No	Class Name	Description
1	<code>HibernateCourseDao</code>	Provides methods to manage course data in database
2	<code>HibernateExamDao</code>	Provides methods to manage test data in database
3	<code>HibernateLessonDao</code>	Provides methods to manage lesson data in database
4	<code>HibernateLessonVersionDao</code>	Provides methods to manage versions of lesson in database
5	<code>HibernateQuestionDao</code>	Provides methods to manage question data in database
6	<code>HibernateReportDao</code>	Provides methods to manage report data in database
7	<code>HibernateUserDao</code>	Provides methods to manage user data in database
8	<code>HibernateAnswerDao</code>	Provides methods to manage course data in database
9	<code>HibernateExamQuestionDao</code>	Provides methods to manage test data in database

4.2.11.3.1 HibernateCourseDao Class

Methods

No	Method	Description
1	save()	Save a course to database
2	findById()	Find a course by its id
3	findByExample()	Find all course have same feature with example
4	findByExample()	Find some course have same feature with example
5	getAll()	Return all course
6	update()	Update a course
7	delete()	Delete a course
8	getCount()	Return the number of the course in database
9	executeSql()	Return the courses execute by sql

4.2.11.3.2 HibernateExamDao Class

Methods

No	Method	Description
1	save()	Save an exam to database
2	findById()	Find an exam by its id
3	findByExample()	Find all exam have same feature with example
4	findByExample()	Find some exam have same feature with example
5	getAll()	Return all exam
6	update()	Update an exam
7	delete()	Delete an exam
8	getCount()	Return the number of the exam in database
9	executeSql()	Return the exams execute by sql

4.2.11.3.3 HibernateLessonDao Class**Methods**

No	Method	Description
1	save()	Save a lesson to database
2	findById()	Find a lesson by its id
3	findByExample()	Find all lesson have same feature with example
4	findByExample()	Find some lesson have same feature with example
5	getAll()	Return all lesson
6	update()	Update a lesson
7	delete()	Delete a lesson
8	getCount()	Return the number of the lesson in database
9	executeSql()	Return the lessons execute by sql

4.2.11.3.4 HibernateLessonVersionDao Class**Methods**

No	Method	Description
1	save()	Save a lesson version to database
2	findById()	Find a lesson version by its id
3	findByExample()	Find all lesson version have same feature with example
4	findByExample()	Find some lesson version have same feature with example
5	getAll()	Return all lesson version
6	update()	Update a lesson version
7	delete()	Delete a lesson version
8	getCount()	Return the number of the lesson version in database

9	executeSql()	Return the lesson versions execute by sql
---	--------------	---

4.2.11.3.5 HibernateQuestionDao Class**Methods**

No	Method	Description
1	save()	Save a question to database
2	findById()	Find a question by its id
3	findByExample()	Find all question have same feature with example
4	findByExample()	Find some question have same feature with example
5	getAll()	Return all question
6	update()	Update a question
7	delete()	Delete a question
8	getCount()	Return the number of the question in database
9	executeSql()	Return the questions execute by sql

4.2.11.3.6 HibernateReportDao Class**Methods**

No	Method	Description
1	save()	Save a report to database
2	findById()	Find a report by its id
3	findByExample()	Find all report have same feature with example
4	findByExample()	Find some report have same feature with example
5	getAll()	Return all report
6	update()	Update a report
7	delete()	Delete a report
8	getCount()	Return the number of the report in database

9	executeSql()	Return the reports execute by sql
---	--------------	-----------------------------------

4.2.11.3.7 HibernateUserDao Class**Methods**

No	Method	Description
1	save()	Save an account to database
2	findById()	Find an account by its id
3	findByExample()	Find all account have same feature with example
4	findByExample()	Find some account have same feature with example
5	getAll()	Return all account
6	update()	Update an account
7	delete()	Delete an account
8	getCount()	Return the number of the account in database
9	executeSql()	Return the accounts execute by sql

4.2.11.3.8 HibernateAnswerDao Class**Methods**

No	Method	Description
1	save()	Save an answer to database
2	findById()	Find an answer by its id
3	findByExample()	Find all answer have same feature with example
4	findByExample()	Find some answer have same feature with example
5	getAll()	Return all answer
6	update()	Update an answer
7	delete()	Delete an answer
8	getCount()	Return the number of the answer in database

9	executeSql()	Return the answer execute by sql
---	--------------	----------------------------------

4.2.11.3.9 HibernateExamQuestionDao Class

Methods

No	Method	Description
1	save()	Save an exam question to database
2	findById()	Find an exam question by its id
3	findByExample()	Find all exam question have same feature with example
4	findByExample()	Find some exam question have same feature with example
5	getAll()	Return all exam question
6	update()	Update an exam question
7	delete()	Delete an exam question
8	getCount()	Return the number of the exam question in database
9	executeSql()	Return the exam questions execute by sql

4.2.11.4 core.exception Package

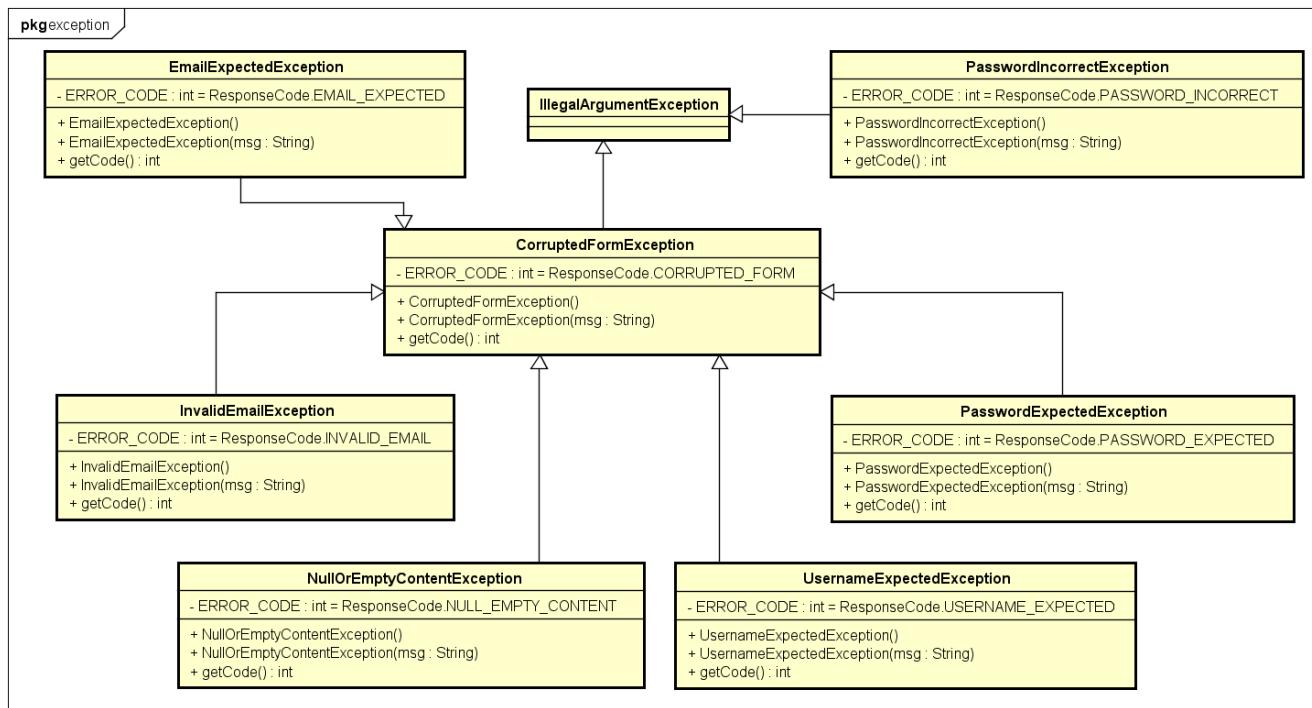


Figure 4: Class diagram Practice core.exception package

No	Class Name	Description
1	CorruptedFormException	Exception throwed when user submitted a corrupted form i.e. form that does not have enough expected data
2	EmailExpectedException	Exception throwed when submitted form does not contain email
3	InvalidEmailException	Exception throwed when submitted form contains invalid email
4	PasswordExpectedException	Exception throwed when submitted form does not contain password
5	PasswordIncorrectException	Exception throwed when submitted form contains incorrect username or password

6	UsernameExpectedException	Exception throwed when submitted form does not contain username
7	NullOrEmptyContentException	Exception throwed when submitted data is null or empty

4.2.11.4.1 CorruptedFormException Class

Attributes

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.2 EmailExpectedException Class

Attributes

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.3 InvalidEmailException Class

Attributes

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.4 PasswordExpectedException Class**Attributes**

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.5 PasswordIncorrectException Class**Attributes**

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.6 UsernameExpectedException Class**Attributes**

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.4.7 NullOrEmptyContentException Class**Attributes**

No	Attribute	Type	Default	Note	Description
1	ERROR_CODE	int		private	

Methods

No	Method	Description
1	getCode()	

4.2.11.5 core.form Package

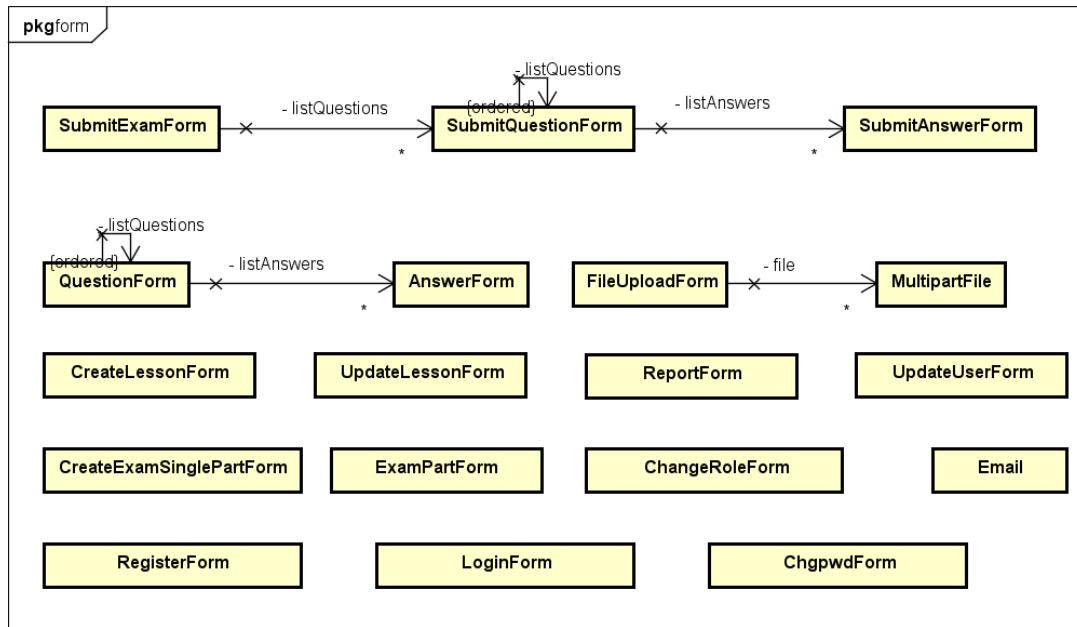


Figure 5: Class diagram `core.form` package

No	Class Name	Description
1	AnswerForm	Answer form
2	ChangeRoleForm	Change user role form
3	ChgpwdForm	Change password form
4	CreateExamSinglePartForm	Create exam form
5	CreateLessonForm	Create lesson form
6	ExamPartForm	Form for generating a exam
7	FileUploadForm	Upload file form
8	LoginForm	Login form Lookup word form
9	QuestionForm	Get a question form
10	RegisterForm	Register form
11	ReportForm	Report a question or lesson form

12	SubmitExamForm	Submit an exam
13	SubmitQuestionForm	Submit questions of an exam
14	SubmitAnswerForm	Submit answers of a question
15	UpdateLessonForm	Update question lesson form
16	UpdateUserForm	Update user form
17	Mail	Send email form

4.2.11.5.1 AnswerForm Class

Attributes

No	Attribute	Type	Note	Description
1	answer	String	private	The question answer.
2	isRight	Boolean	private	The answer is correct or not

4.2.11.5.2 ChangeRoleForm Class

Attributes

No	Attribute	Type	Note	Description
1	userId	Integer	private	The user id.
2	role	Integer	private	The new role of user.

4.2.11.5.3 ChgpwdForm Class

Attributes

No	Attribute	Type	Note	Description
1	oldPassword	String	private	Old password.
2	newPassword	String	private	New password.

4.2.11.5.4 CreateExamSinglePartForm Class

Attributes

No	Attribute	Type	Note	Description
1	courseId	Integer	private	The course of the exam
2	skill	Integer	private	The skill of the exam

3	numberOfQuestion	Integer	private	The number question of the exam
---	------------------	---------	---------	---------------------------------

4.2.11.5.5 CreateLessonForm Class

Attributes

No	Attribute	Type	Note	Description
1	courseld	Integer	private	The course id of lesson.
2	lessonTitle	String	private	The lesson title.
3	description	String	private	The lesson description.
4	vocabulary	String	private	The lesson vocabulary.
5	grammar	String	private	The lesson grammar.
6	reading	String	private	The lesson reading part.
7	listening	String	private	The lesson listening part.
8	practice	String	private	The lesson practice part.
9	article	String	private	The lesson article.

4.2.11.5.6 ExamPartForm Class

Attributes

No	Attribute	Type	Note	Description
1	skill	int	private	The skill of exam.
2	numberOfQuestion	int	private	The number of question in exam.

4.2.11.5.7 FileUploadForm Class

4.2.11.5.8 LoginForm Class

Attributes

No	Attribute	Type	Note	Description
1	username	String	private	The user name.
2	encryptedPassword	String	private	The user password has been encrypted.

4.2.11.5.9 QuestionForm Class**Attributes**

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question Id.
2	questionAnswerType	Integer	private	The question answer type.
3	questionType	Integer	private	The question type.
4	questionSkill	Integer	private	The question skill.
5	courseId	Integer	private	The course id of question.
6	creatorId	Integer	private	The id of user who create question.
7	numberOfQuestion	Integer	private	The number of child question.
8	etaTime	Integer	private	Estimate time to complete the question
9	question	String	private	The question.
10	attachment	String	private	The file attach with question

4.2.11.5.10 RegisterForm Class**Attributes**

No	Attribute	Type	Note	Description
1	username	String	private	The user name.
2	password	String	private	The user password.
3	email	String	private	The user email.

4.2.11.5.11 ReportForm Class**Attributes**

No	Attribute	Type	Note	Description
1	content	String	private	The report content.

4.2.11.5.12 SubmitExamForm Class

Attributes

No	Attribute	Type	Note	Description
1	examId	Integer	private	The exam id.
2	takenTime	Integer	private	The time user do exam.

4.2.11.5.13 SubmitQuestionForm Class

Attributes

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question id.

4.2.11.5.14 SubmitAnswerForm Class

Attributes

No	Attribute	Type	Note	Description
1	answer	String	private	The answer
2	isSelected	Boolean	private	User select this answer or not

4.2.11.5.15 UpdateLessonForm Class

Attributes

No	Attribute	Type	Note	Description
1	lessonId	Integer	private	The lesson id
2	courseld	Integer	private	The course id of lesson.
3	lessonTitle	String	private	The lesson title.
4	description	String	private	The lesson description.
5	vocabulary	String	private	The lesson vocabulary.
6	grammar	String	private	The lesson grammar.
7	reading	String	private	The lesson reading part.
8	listening	String	private	The lesson listening part.
9	practice	String	private	The lesson practice part.
10	article	String	private	The lesson article.

4.2.11.5.16 UpdateUserForm Class**Attributes**

No	Attribute	Type	Note	Description
1	firstName	String	private	The user first name.
2	lastName	String	private	The user last name.
3	dob	Long	private	date of birth
4	address	String	private	The user address.
5	hobby	String	private	The user hobby.
6	bio	String	private	The user introduction.
7	website	String	private	The user website.
8	quote	String	private	Favorite quote

4.2.11.5.17 Email Class**Attributes**

No	Attribute	Type	Note	Description
1	senderEmailAddress	String	private	Email address of sender
2	senderPassword	String	private	Password of sender
3	receiverEmailAddress	String	private	Email address of receiver
4	emailSubject	String	private	Email subject
5	emailContent	String	private	Email content
6	emailContentType	String	private	Email content type

4.2.11.6 core.model package

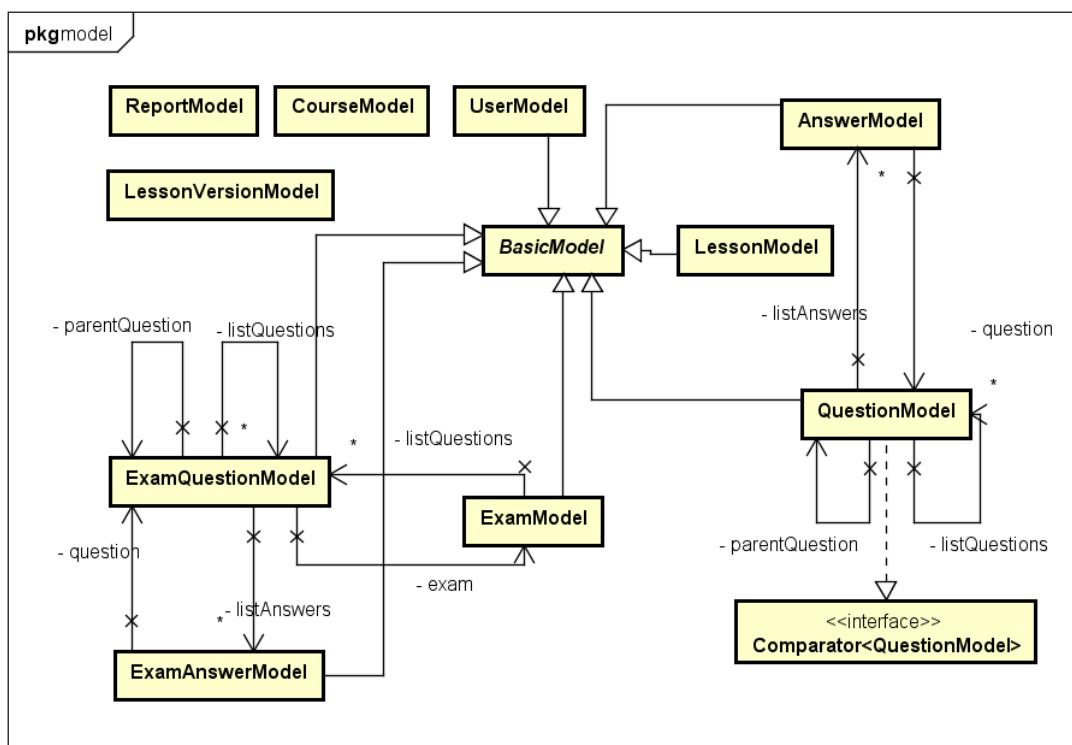


Figure 6: Class diagram core.model package

No	Class Name	Description
1	<code>AnswerModel</code>	Answer model
2	<code>BasicModel</code>	Basic model
3	<code>CourseModel</code>	Course model
4	<code>ExamAnswerModel</code>	Exam answer
5	<code>ExamModel</code>	Exam model
6	<code>ExamQuestionModel</code>	Exam question model
7	<code>LessonModel</code>	Lesson model
8	<code>LessonVersionModel</code>	Lesson version model
9	<code>QuestionModel</code>	Question model
10	<code>ReportModel</code>	Report model
11	<code>UserModel</code>	User model

4.2.11.6.1 AnswerModel Class

Attributes

No	Attribute	Type	Note	Description
1	answer	String	private	The question answer.
2	isRight	Boolean	private	The answer is correct or not

4.2.11.6.2 BasicModel Class

Attributes

No	Attribute	Type	Note	Description
1	id	Integer	private	The model id
2	createDate	Long	private	The date model is created
3	updateDate	Long	private	The last update date
4	deleteDate	Long	private	The date model is deleted
5	deleteFlag	Boolean	private	The model is deleted or not

4.2.11.6.3 CourseModel Class

Attributes

No	Attribute	Type	Note	Description
1	id	Integer	private	The course id
2	name	String	private	The course name
3	description	String	private	The course description

4.2.11.6.4 ExamAnswer Class

Attributes

No	Attribute	Type	Default	Note	Description
1	answer	String		private	The answer.
2	isRight	Boolean		private	The answer is correct or not.
3	isSelected	Boolean		private	The answer is selected by learner or not.

4.2.11.6.5 ExamModel Class

Attributes

No	Attribute	Type	Note	Description

1	userId	Integer	private	The id of user who do exam
2	courseld	Integer	private	The course id of exam
3	questionSkill	Integer	private	The skill of exam
4	result	Double	private	The result
5	takenTime	Integer	private	The time user do exam
6	etaTime	Integer	private	The total time of exam
7	finishState	Boolean	private	The exam has been finished or not

4.2.11.6.6 ExamQuestionModel Class**Attributes**

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question id
2	question	String	private	The question
3	questionType	Integer	private	The question type
4	numberOfQuestion	Integer	private	The number of child question
5	questionAnswerType	Integer	private	The question answer type
6	attachment	String	private	The file attach with question
7	isChanged	Boolean	private	The question has been changed or not

4.2.11.6.7 LessonModel Class**Attributes**

No	Attribute	Type	Note	Description
1	index	Integer	private	The index of lesson in course.
2	courseld	Integer	private	The course id of lesson.
3	currentVersionId	Integer	private	The current lesson version id.

4.2.11.6.8 LessonVersionModel Class**Attributes**

No	Attribute	Type	Note	Description
----	-----------	------	------	-------------

1	id	Integer	private	The lesson version id
2	createDate	Long	private	The create date.
3	updateDate	Long	private	The update date
4	lessonId	Integer	private	The lesson id.
5	version	Integer	private	The lesson version
6	state	Integer	private	The state of lesson version
7	title	String	private	The lesson title.
8	description	String	private	The lesson description.
9	creatorId	Integer	private	The creator id
10	vocabulary	String	private	The lesson vocabulary.
11	grammar	String	private	The lesson grammar.
12	reading	String	private	The lesson reading part.
13	listening	String	private	The lesson listening part.
14	practice	String	private	The lesson practice part.
15	article	String	private	The lesson article.

4.2.11.6.9 QuestionModel Class**Attributes**

No	Attribute	Type	Note	Description
1	questionCode	Integer	private	The question code
2	creatorId	Integer	private	The creator id
3	questionAnswerType	Integer	private	The question answer type
4	questionType	Integer	private	The question type
5	questionSkill	Integer	private	The question skill
6	numberOfQuestion	Integer	private	The number of child questions
7	courseld	Integer	private	The course id

8	question	String	private	The question
9	content	List<Integer>	private	The child questions id
10	state	Integer	private	The question state
11	attachment	Integer	private	The file url attach with question

4.2.11.6.10 ReportModel Class

Attributes

No	Attribute	Type	Note	Description
1	id	Integer	private	The report id
2	createDate	Long	private	The create date
3	deleteDate	Long	private	The delete date
4	deleteFlag	Long	private	The report is deleted or not
5	senderId	Integer	private	The user who send report
6	receiverId	Integer	private	The user who receive report
7	readFlag	Boolean	private	Check if report is read or not
8	content	String	private	The report content
9	lessonId	Integer	private	The id of lesson is reported
10	questionId	Integer	private	The id of question is reported

4.2.11.6.11 UserModel Class

Attributes

No	Attribute	Type	Note	Description
1	email	String	private	The user email
2	userName	String	private	The user name
3	encryptedPassword	String	private	The password had been encrypted
4	firstName	String	private	The user first name.
5	lastName	String	private	The user last name.

6	dob	Long	private	Date of birth
7	joinDate	Long	private	The join date
8	address	String	private	The user address.
9	role	Integer	private	The user role
10	hobby	String	private	The user hobby.
11	bio	String	private	The user introduction.
12	website	String	private	The user website.
13	quote	String	private	Favorite quote
14	isBanned	Boolean	private	The user is banned or not

4.2.11.7 core.response Package

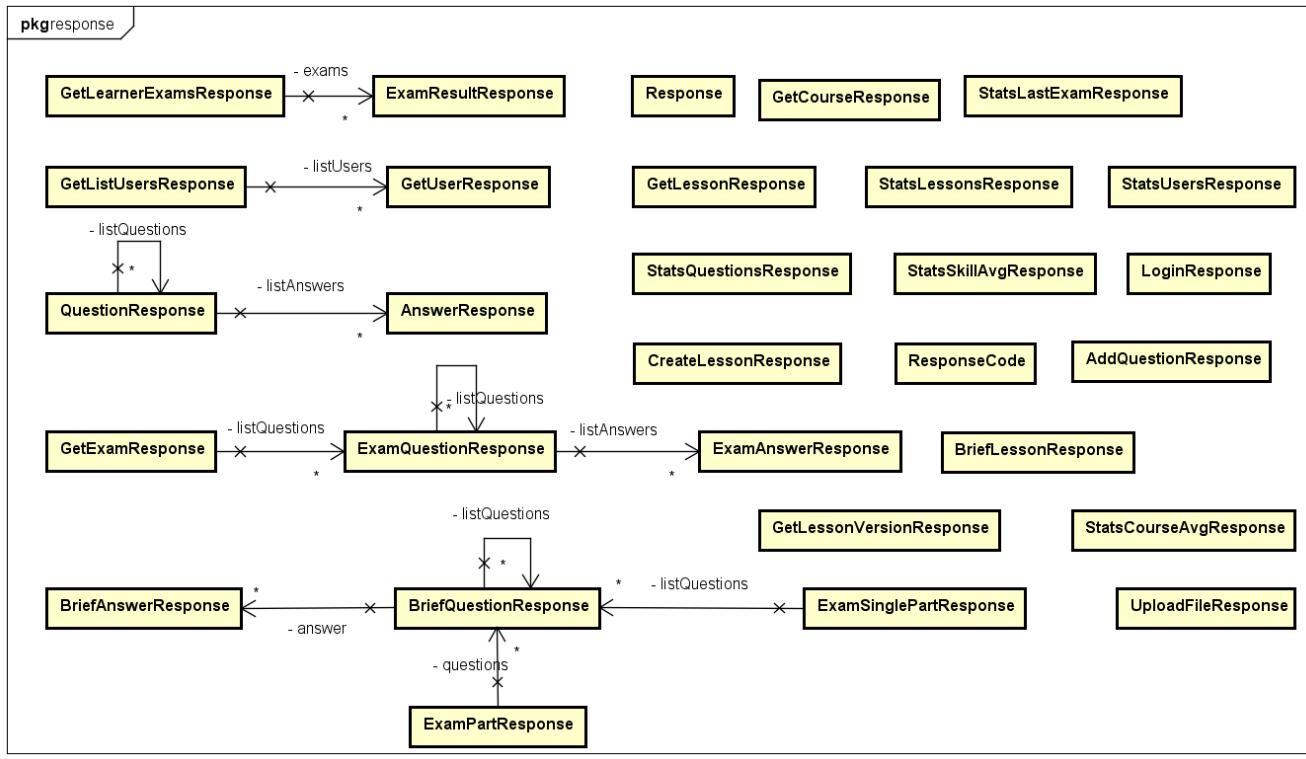


Figure 7: Class diagram core.response package

No	Class Name	Description
1	AddQuestionResponse	Response for add question API
2	AnswerResponse	Answer of a question
3	BriefAnswerResponse	Brief of an answer
4	BriefLessonResponse	Brief of an lesson
5	BriefQuestionResponse	Brief of an question
6	CreateLessonResponse	Response for create lesson API
7	ExamAnswerResponse	Answer of a question in exam
8	ExamPartResponse	Response for a generated an exam
9	ExamQuestionResponse	Question in exam
10	ExamResultResponse	Result of exam

11	ExamSinglePartResponse	Response for generating a single-skill exam
12	GetCourseResponse	Response for get course API
13	GetExamResponse	Response for get exam API
14	GetLearnerExamsResponse	Response for get learner exam API
15	GetLessonResponse	Response for get lesson API
16	GetLessonVersionResponse	Response for get lesson version API
17	GetListUsersResponse	Response for get list users API
18	GetUserResponse	Response for get a user API
19	LoginResponse	Response for login API
20	QuestionResponse	Contain question content
21	Response	Response form for all API
22	ResponseCode	Return code for each API
23	StatsCourseAvgResponse	Response for get statistic of exam in a course API
24	StatsLastExamResponse	Response for get statistic of last exam API
25	StatsLessonsResponse	Response for get statistic of lesson API
26	StatsQuestionsResponse	Response for get statistic of question API
27	StatsSkillAvgResponse	Response for get statistic of exam in a skill API
28	StatsUsersResponse	Response for get statistic of an user API
29	UploadFileResponse	Response for upload file API

4.2.11.7.1 AddQuestionResponseData Class**Attributes**

No	Attribute	Type	Default	Note	Description
1	questionId	Integer		private	question id

4.2.11.7.2 AnswerResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	answer	String	private	The question answer.
2	isRight	Boolean	private	The answer is correct or not

4.2.11.7.3 BriefAnswerResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	answer	String	private	The question answer.
2	isRight	Boolean	private	The answer is correct or not
3	isSelected	Boolean	private	User choose answer or not

4.2.11.7.4 BriefLessonResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	lessonId	Integer	private	The lesson id
2	title	String	private	The lesson type
3	state	Integer	private	The lesson state
4	courseld	Integer	private	The course id
5	version	Integer	private	The lesson version
6	description	String	private	The lesson description

4.2.11.7.5 BriefQuestionResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question id

2	question	String	private	The question
3	attachment	String	private	The file url attach with question

4.2.11.7.6 CreateLessonResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	lessonId	Integer	private	The lesson version
2	courseld	Integer	private	The state of lesson version
3	version	Integer	private	The lesson version
4	lessonTitle	String	private	The lesson title.
5	description	String	private	The lesson description.
6	creatorId	Integer	private	The creator id
7	createDate	Long	private	The creation date
8	vocabulary	String	private	The lesson vocabulary.
9	grammar	String	private	The lesson grammar.
10	reading	String	private	The lesson reading part.
11	listening	String	private	The lesson listening part.
12	practice	String	private	The lesson practice part.
13	article	String	private	The lesson article.

4.2.11.7.7 ExamAnswerResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	answer	Integer	private	The question answer.
2	isRight	Boolean	private	The answer is correct or not
3	isSelected	Boolean	private	User choose answer or not

4.2.11.7.8 ExamPartResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	examId	Integer	private	The exam id
2	skill	Integer	private	The exam skill
3	etaTime	Integer	private	The total time of exam
4	offlineCheck	Boolean	private	The backend side check exam result or not

4.2.11.7.9 ExamQuestionResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question id
2	questionType	Integer	private	The question type
3	question	String	private	The question
4	numberOfQuestion	Integer	private	The number of child questions
5	questionAnswerType	Integer	private	The question answer type
6	attachment	String	private	The file url attach with question
7	isChanged	Boolean	private	The question has been changed or not

4.2.11.7.10 ExamResultResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	examId	Integer	private	The exam id
2	course	Integer	private	The course id
3	skill	Integer	private	The exam skill

4	offlineCheck	Boolean	private	The backend side check exam result or not
5	etaTime	Integer	private	The total time of exam

4.2.11.7.11 ExamSinglePartResponseClass

Attributes

No	Attribute	Type	Note	Description
1	examId	Integer	private	The exam id
2	result	Double	private	The exam result

4.2.11.7.12 GetCourseResponse Class

Attributes

No	Attribute	Type	Default	Note	Description
1	name	String		private	The course name
2	description	String		private	The course desction

4.2.11.7.13 GetExamResponse Class

Attributes

No	Attribute	Type	Note	Description
1	examId	Integer	private	The exam id
2	courseld	Integer	private	The course id
3	questionSkill	Integer	private	The question skill
4	result	Double	private	The exam result
5	takenTime	Integer	private	The time user do exam
6	etaTime	Integer	private	The total time of exam

4.2.11.7.14 GetLearnerExamsresponseData Class

Attributes

No	Attribute	Type	Note	Description
1	exams	ExamResponseData	private	Exam response data

4.2.11.7.15 GetLessonResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	lessonId	Integer	private	The lesson id
2	courseld	Integer	private	The course id
3	lessonTitle	String	private	The lesson title
4	description	String	private	The lesson description
5	updateDate	Long	private	The update date
6	state	Integer	private	The lesson state
7	version	Integer	private	The lesson version
8	vocabulary	String	private	The lesson vocabulary.
9	grammar	String	private	The lesson grammar.
10	conversation	String	private	The lesson conversation part.
11	listening	String	private	The lesson listening part.
12	practice	String	private	The lesson practice part.
13	reading	String	private	The lesson reading.
14	previousLessonId	Integer	private	The previous lesson on same course
15	nextLessonId	Integer	private	The next lesson on same course

4.2.11.7.16 GetLessonVersionResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	lessonId	Integer	private	The lesson id.
2	courseld	Integer	private	The lesson version
3	version	Integer	private	The lesson version

4	lessonTitle	String	private	The lesson title
5	description	String	private	The lesson description
6	creatorId	Integer	private	The creator id
7	createDate	Long	private	The creation date.
8	updateDate	Long	private	The update date
9	vocabulary	String	private	The lesson vocabulary.
10	grammar	String	private	The lesson grammar.
11	reading	String	private	The lesson reading part.
12	listening	String	private	The lesson listening part.
13	practice	String	private	The lesson practice part.
14	article	String	private	The lesson article.
15	listVersions	List< Integer >	private	List lesson version id in same lesson

4.2.11.7.17 GetListUsersResponseData Class**Attributes**

No	Attribute	Type	Note	Description
1	listUsers	List< GetUserResponseData >	private	

4.2.11.7.18 GetUserResponseData Class**Attributes**

No	Attribute	Type	Note	Description
1	email	String	private	The user email
2	userName	String	private	The user name
3	firstName	String	private	The user first name.
4	lastName	String	private	The user last name.
5	dob	Long	private	Date of birth
6	joinDate	Long	private	The join date
7	address	String	private	The user address.

8	role	Integer	private	The user role
9	hobby	String	private	The user hobby.
10	bio	String	private	The user introduction.
11	website	String	private	The user website.
12	quote	String	private	Favorite quote

4.2.11.7.19 LoginResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	role	Integer	private	The role of user

4.2.11.7.20 QuestionResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	questionId	Integer	private	The question id
2	questionCode	Integer	private	The question code
3	questionAnswerType	Integer	private	The question answer type
4	questionType	Integer	private	The question type
5	questionSkill	Integer	private	The question skill
6	courseld	Integer	private	The course id
7	question	String	private	The question
8	state	Integer	private	The question state
9	createDate	Long	private	The creation date
10	updateDate	Long	private	The update date

4.2.11.7.21 Response Class**Attributes**

No	Attribute	Type	Note	Description
1	code	Integer	private	Response code.
2	data	String	private	Response detail content.

4.2.11.7.22 ResponseCode Class**Attributes**

No	Attribute	Type	Value
1	SUCCESS	int	200
2	CREATED	int	201
3	FOUND	int	302
4	BAD_REQUEST	int	400
5	UNAUTHORIZED	int	401
6	PERMISSION_DENIED	int	403
7	REQUEST_TIMEOUT	int	408
8	INVALID_EMAIL	int	410
9	DUPLICATED_EMAIL	int	411
10	EMAIL_NOT_FOUND	int	412
11	DUPLICATED_USERNAME	int	413
12	CORRUPTED_FORM	int	460
13	USERNAME_EXPECTED	int	461
14	PASSWORD_EXPECTED	int	462
15	EMAIL_EXPECTED	int	463
16	INVALID_PASSWORD	int	420
17	INCORRECT_PASSWORD	int	421
18	USER_NOT_FOUND	int	430
19	USER_NOT_ALLOW	int	431
20	USER_INACTIVE	int	432
21	USER_EXAMS_NOT_FOUND	int	433
22	USERNAME_PASSWORD_NOT_MATCH	int	470
23	PASSWORD_INCORRECT	int	471

24	QUESTION_NOT_FOUND	int	440
25	INTERNAL_SERVER_ERROR	int	500
26	UNSUPPORTED_API	int	600

4.2.11.7.23 StatsLastExamResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	result	Double	private	Last exam result

4.2.11.7.24 StatsLessonsResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	numberOfLessons	Integer	private	The number of lesson in system.

4.2.11.7.25 StatsQuestionsResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	numberOfLessons	Integer	private	The number of question in system.

4.2.11.7.26 StatsSkillAvgResponse Class**Attributes**

No	Attribute	Type	Note	Description
1	listening	Double	private	Average point in listening skills.
2	reading	Double	private	Average point in reading skills.
3	vocabulary	Double	private	Average point in vocabulary skills.
4	grammar	Double	private	Average point in grammar skills.

4.2.11.7.27 StatsUsersResponse Class

Attributes

No	Attribute	Type	Note	Description
1	activeUser	Integer	private	The number of user who isn't banned
2	totalUser	Integer	private	The number of user in system

4.2.11.7.28 UploadFileResponse Class

Attributes

No	Attribute	Type	Note	Description
1	link	String	private	Uploaded file url.

4.2.11.8 core.security package

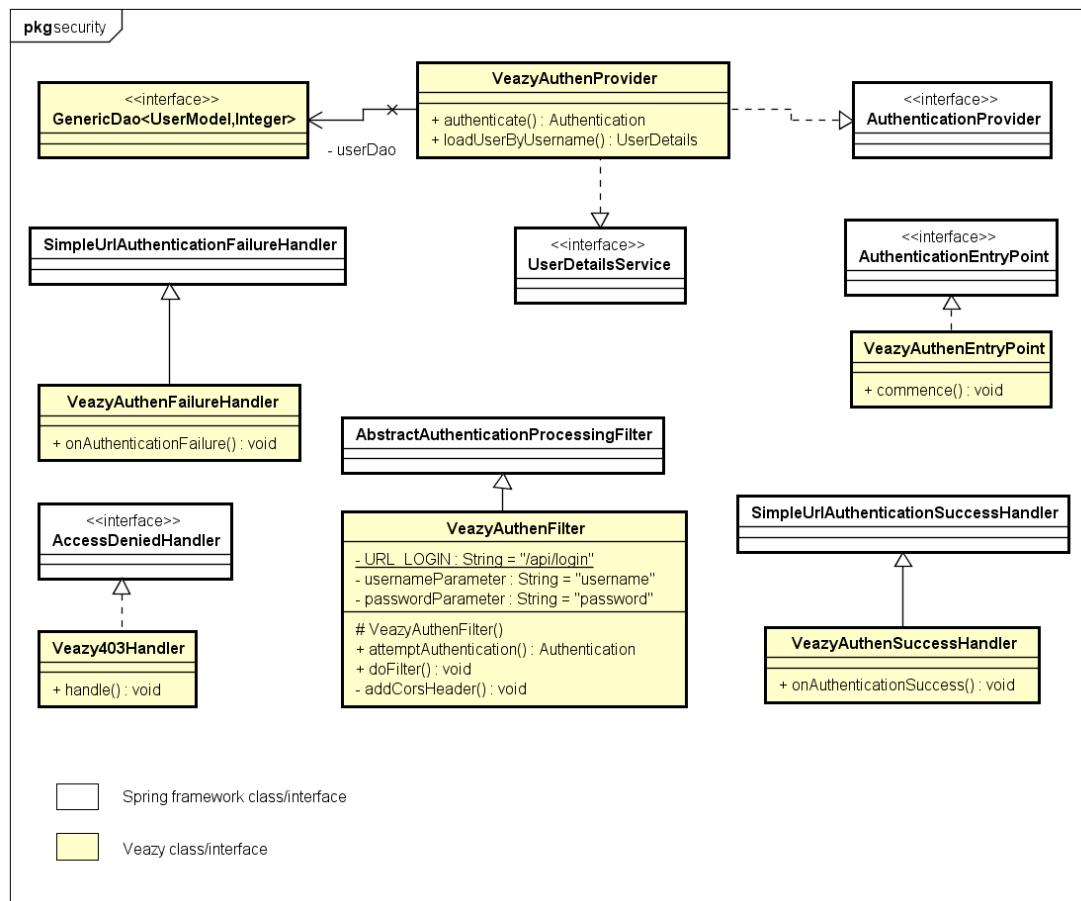


Figure 8: Class diagram `core.security` package

No	Class Name	Description
1	<code>Veazy403Handler</code>	Handles 403 forbidden error
2	<code>VeazyAuthenEntryPoint</code>	Handles anonymous requests
3	<code>VeazyAuthenFailureHandler</code>	Handles authentication failure requests
4	<code>VeazyAuthenFilter</code>	Filters authentication requests
5	<code>VeazyAuthenProvider</code>	Provides authentication mechanism
6	<code>VeazyAuthenSuccessHandler</code>	Handles authentication success requests

4.2.11.8.1 `Veazy403Handler` Class

Methods

No	Method	Description
1	<code>handle()</code>	Handles denied request

4.2.11.8.2 VeazyAuthenEntryPoint Class**Methods**

No	Method	Description
1	commence()	Handles anonymous request

4.2.11.8.3 VeazyAuthenFailureHandler Class**Methods**

No	Method	Description
1	onAuthenticationFailure()	Handles failed authentication event

4.2.11.8.4 VeazyAuthenFilter Class**Attributes**

No	Attribute	Type	Note	Description
1	URL_LOGIN	String	private	Defines url string to filter the authentication request
2	usernameParameter	String	private	Defines the username parameter for UsernamePasswordAuthen tication
3	passwordParameter	String	private	Defines the password parameter for UsernamePasswordAuthen tication

Methods

No	Method	Description
1	attemptAuthentication()	Attempts to authenticate the requested principal
2	doFilter()	Filters the authentication request
3	addCorsHeader()	Adds CORS header to response if needed

4.2.11.8.5 VeazyAuthenProvider Class

Methods

No	Method	Description
1	authenticate()	Authenticates the requested principal
2	loadUserByUsername()	Loads user details from requested principal

4.2.11.8.6 VeazyAuthenSuccessHandler Class

Methods

No	Method	Description
1	onAuthenticationSuccess()	Handles successful authentication event

4.2.11.9 core.service Package

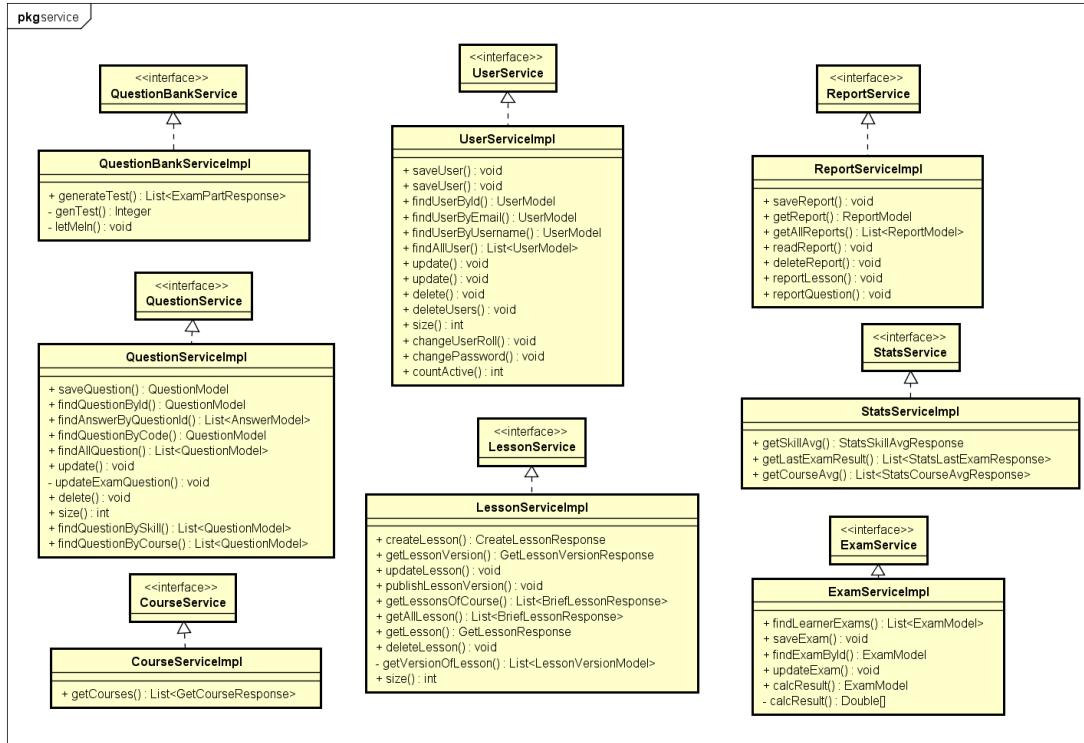


Figure 9: Class diagram core.service package

No	Class Name	Description
1	CourseServiceImpl	Contains business logical functions for processing course data
2	ExamServiceImpl	Contains business logical functions for processing test data
3	LessonServiceImpl	Contains business logical functions for processing lesson data
4	QuestionBankServiceImpl	Contains business logical functions for processing question bank data
5	QuestionServiceImpl	Contains business logical functions for processing question data
6	ReportServiceImpl	Contains business logical functions for processing report data
7	StatsServiceImpl	Contains business logical functions for processing statistic data

8	UserServiceImpl	Contains business logical functions for processing user data
---	-----------------	--

4.2.11.9.1 CourseServiceImpl Class**Methods**

No	Method	Description
1	getCourses()	Return list course

4.2.11.9.2 ExamServiceImpl Class**Methods**

No	Method	Description
1	findLearnerExams()	Return all done exam of a user
2	saveExam()	Save an exam to database
3	findExamById()	Return an exam by its id
4	updateExam()	Update exam to database
5	calcResult()	Calculate exam result

4.2.11.9.3 LessonServiceImpl Class**Methods**

No	Method	Description
1	createLesson()	Create new lesson
2	getLessonVersion()	Get a version of lesson
3	updateLesson()	Update a lesson
4	publishLessonVersion()	Make lesson as published version. Learner can see this lesson.
5	getLessonsOfCourse()	Return list lesson of a course.
6	getLesson()	Return a lesson.
7	deleteLesson()	Delete a lesson.
8	getAllLesson()	Return all lesson.

4.2.11.9.4 QuestionBankServiceImpl Class Methods

No	Method	Description
1	generateExam()	Generates the exam response from the list of random questions
2	genExam()	Generates list of random questions
3	letMeIn()	Puts questions details to suitable list

4.2.11.9.5 QuestionServiceImpl Class Methods

No	Method	Description
1	saveQuestion()	Save a question
2	findQuestionById()	Find a question by its id.
3	findAnswerByQuestionId()	Find all answer of a question by question id
4	findQuestionByCode()	Find a question by question code.
5	findAllQuestion()	Get all question.
6	update()	Update a question.
7	updateExamQuestion()	Update information of exam question what link to updated question.
8	delete()	Delete a question.
9	size()	Return the number of question in database.
10	findQuestionBySkill()	Find all question by a skill
11	findQuestionByCourse()	Find all question in a course

4.2.11.9.6 ReportServiceImpl Class Methods

No	Method	Description
1	saveReport()	Save a report to database.
2	getReport()	Get a report.
3	getAllReports()	Get all report.

4	<code>readReport()</code>	Make a report as read state.
5	<code>deleteReport()</code>	Delete a report.
6	<code>reportLesson()</code>	User report a lesson.
7	<code>reportQuestion()</code>	User report a question

4.2.11.9.7 StatsServiceImpl Class Methods

No	Method	Description
1	<code>getSkillAvg()</code>	Return average exam point for each skill of a user.
2	<code>getLastExamResult()</code>	Return last exam result of a user.
3	<code>getCourseAvg()</code>	Return average exam point for each course of a user.

4.2.11.9.8 UserServiceImpl Class Methods

No	Method	Description
1	<code>saveUser</code>	Save new user.
2	<code>findUserById</code>	Find a user by id
3	<code>findUserByEmail</code>	Find a user by email
4	<code>findUserByUsername</code>	Find a user by user name
5	<code>findAllUser</code>	Get all user.
6	<code>update</code>	Update a user.
7	<code>delete</code>	Delete a user.
9	<code>deleteUsers</code>	Delete multiple user
10	<code>size</code>	Return the number of user in system.
11	<code>changeUserRole</code>	Change user role
12	<code>changePassword</code>	Change user password
13	<code>countActive</code>	Return the number of active user

4.3 Screen Design

4.3.1 Screen Flow

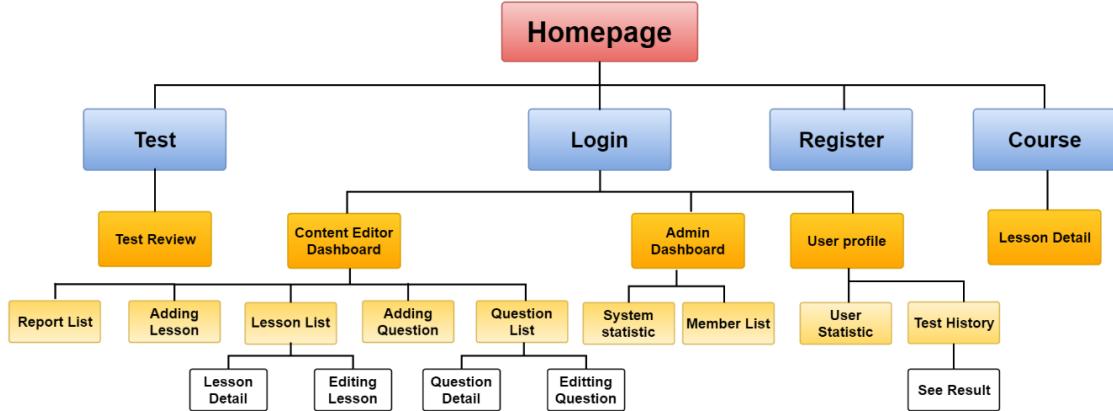


Figure 4-36: Screen Flow

No	Screen Name	Description
1	Homepage	Homepage
2	Login	Logging into Veazy system with a registered account
3	Register	Registering new account
4	Courses	Displaying list of courses and available lessons in each course
5	Lesson Detail	Displaying lesson content
6	Test	Setting up test
7	Test Review	Reviewing test
8	User Profile	Displaying detailed information of user
9	User Statistic	Displaying statistics of user
10	Test History	Displaying list of taken tests
11	See Result	Displaying content of taken test
12	Report List	Viewing list report
13	Lesson List (for content manager & content creator)	Displaying lesson list
14	Lesson Detail (for content manager & content creator)	Displaying lesson detail
15	Adding Lesson	Creating new lesson
16	Editing Lesson	Editing lesson content

17	Question List	Displaying list of questions intended to be generated in test
18	Question Detail	Displaying question content
19	Adding Question	Creating new question
20	Editing Question	Editing question content
21	System Statistic	Displaying system statistics
22	Member List	Displaying list of Veazy members
23	Test History	Displaying list test history of member

4.3.2 Guest

4.3.2.1 Homepage

The screenshot shows the Veazy homepage. At the top, there is a navigation bar with links for Home, Courses, and Test, along with a Login button. The main header features the word "VEAZY" in orange with a small "JA" suffix. Below the header is a large banner with the text "Welcome to Veazy!" and a "Sign up" button. The banner background is a photograph of an open book. Underneath the banner, there is a section titled "Veazy?" with a descriptive paragraph about the platform's purpose. To the right of this text is a collage image featuring a map, a clock, a pen, and other study-related items. The next section, "Veazy's Features", lists four main features: Courses, Exam, Dictionary, and Study Statistics, each accompanied by a small icon and a brief description. Below this is a section titled "Veazy Team" which displays eight Polaroid-style photos of a person working at a desk. The footer contains three columns: "Location" with placeholder text, "Around the Web" with social media icons, and "About" with placeholder text. The footer also includes a copyright notice: "Copyright © Veazy Company".

Figure 22. “Home” page

Field name	Type	Mandatory	Min	Max	Description
------------	------	-----------	-----	-----	-------------

			length	length	
Veazy	Hyperlink	N			Hyperlink that change language of website
Home	Hyperlink	N			Hyperlink that redirects to “Home” page
Courses	Select box	N			Select box that redirects to “Courses” page
Test	Hyperlink	N			Hyperlink that redirects to “Test” page
Dictionary	Hyperlink	N			Hyperlink that redirects to “Dictionary” page
Login	Hyperlink	N			Hyperlink that redirects to “Login” page
Sign Up	Hyperlink	N			Button for registering new account

Table 2. Components of “Home” page

4.3.2.2 Login

Veazy

Username

Password

Forgot your password?

Login

Don't have account yet? Click here to register.

Back to Homepage

Figure 4. Login page

Field name	Type	Mandatory	Min length	Max length	Description
Username	Textbox	Y	6	30	Textbox for entering username
Password	Textbox	Y	6	32	Textbox for entering password
Login	Button	Y			Button for submitting login form

Forgot your password?	Hyperlink	N			Hyperlink that redirects to “Password Recover” page
Click here to register.	Hyperlink	N			Hyperlink that redirects to “Register” page
Back to Home	Hyperlink	N			Hyperlink that redirects to “Home” page

Table 3. Components of “Login” page

4.3.2.3 Register

The screenshot shows the registration form of the Veazy application. At the top, there is a logo consisting of an orange paw print icon followed by the word "Veazy" in a stylized font. Below the logo is a circular placeholder for a user profile picture. The form contains four input fields: "Username", "Email", "Password", and "Confirm password". At the bottom left of the form, there is a link "Back to Login page" with a back arrow icon. On the right side, there is a green rectangular button with the word "Register" in white. At the very bottom center of the entire page, there is another link "Back to Homepage".

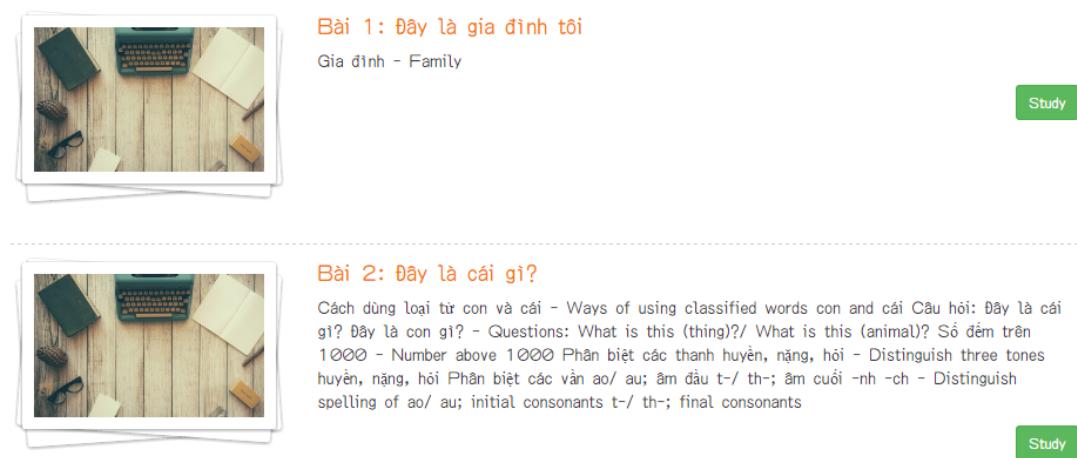
Figure 5. “Register” page

Field name	Type	Mandatory	Min length	Max length	Description
Username	Textbox	Y	8	30	Textbox for entering username
Email	Textbox	Y	6	254	Textbox for entering email
Password	Textbox	Y	8	32	Textbox for entering password
Confirm Password	Textbox	Y	8	32	Textbox for re-entering password
Register	Button	Y			Button to submit register form
Back to Login page	Hyperlink	N			Hyperlink that redirects to “Login” page
Back to Homepage	Hyperlink	N			Hyperlink that redirects to “Home” page

Table 4. Components of “Register” page

4.3.2.4 Courses

Lessons in Vietnamese Beginner Course



Bài 1: Đây là gia đình tôi
Gia đình - Family Study

Bài 2: Đây là cái gì?
Cách dùng loại từ con và cái - Ways of using classified words con and cái Câu hỏi: Đây là cái gì? Đây là con gì? - Questions: What is this (thing)?/ What is this (animal)? Số đếm trên 1000 - Number above 1000 Phân biệt các thanh huyền, nặng, hỏi - Distinguish three tones huyền, nặng, hỏi Phân biệt các vần ao/ au; âm đầu t-/ th-; âm cuối -nh -ch - Distinguish spelling of ao/ au; initial consonants t-/ th-; final consonants Study

Figure 6. “Courses” page

Field name	Type	Mandatory	Min length	Max length	Description
Study	Button	N			Button that redirects to “Lesson” page

Table 5. Course page’s components

4.3.2.5 Lesson

Bài 1: Ngày mai trời thế nào?

Vocab
Grammar
Convo
Listening
Practice
Reading

- "Trời thế nào?" / "Thời tiết thế nào?" dùng để hỏi về thời tiết - "Trời thế nào?" / "Thời tiết thế nào?" is used to ask about weather.
- "Sắp...chưa" dùng để hỏi hỏi về thi tương lai gần - "Sắp...chưa" is used to form the near future tense question,
- Trong câu trả lời thường có từ "rồi" ở cuối câu - "rồi" is often placed at the end of the answer.

- Chưa, tuần sau, ông ấy mới đi
Ông ấy sắp đi Huế chưa?
Ví dụ:
- We use "sẽ" or "mới" for future with specific time.
- We use "sắp" for near future.

Notes: To answer for near future tense question:
Nếu câu trả lời là thi tương lai, với thời gian cụ thể thì phải dùng "sẽ" hoặc "mới":
Chú ý: Khi hỏi về thi tương lai gần "sắp... chưa?", nếu câu trả lời là ở tương lai gần thì chúng ta dùng từ "sắp".

Provided by

Lesson Detail

Course: Upper-beginner
About lesson:
Một số từ ngữ về thời tiết - Number of words about weather
Câu hỏi về thời tiết: Trời thế nào?
- Question about weather:
What's the weather like? Thi
tương lai gần - Near future tense
Câu hỏi sắp ... chưa - Question:
sắp ... chưa

Previous Lesson
Next Lesson

Figure 7. “Lesson Detail” page

Field name	Type	Mandatory	Min length	Max length	Description
Vocab	Hyperlink	N			Hyperlink that displays content of Vocabulary section
Grammar	Hyperlink	N			Hyperlink that displays content of Grammar section
Convo	Hyperlink	N			Hyperlink that displays content of Conversation section
Listening	Hyperlink	N			Hyperlink that displays content of Listening section

Practice	Hyperlink	N			Hyperlink that displays content of Practice section
Reading	Hyperlink	N			Hyperlink that displays content of Reading section
Previous lesson	Hyperlink	N			Hyperlink that redirects to “Lesson” page which displays content of the previous lesson
Next lesson	Hyperlink	N			Hyperlink that redirects to “Lesson” page which displays content of the next lesson
Report	Hyperlink	N			Hyperlink to display lesson report dialog

Table 6. Components of “Lesson” page

4.3.2.6 Test

The screenshot shows a form titled "Test Preparation". It contains four input fields: "Level" (dropdown menu set to "Beginner"), "Test Skill" (dropdown menu set to "Vocabulary"), and "Number of Questions" (text input field containing "1"). Below these fields is a blue "Create" button.

Figure 8. “Test” page

Field name	Type	Mandatory	Min length	Max length	Description
Level	Select box	Y			Select box for selecting difficulty level of test
Test Skill	Select box	Y			Select box for selecting skill to be tested
Question Limit	Number	Y			Textbox for entering the number of questions
Create	Button	Y			Button for submitting test setup form and redirecting to “Take Test” page

Table 7. Components of “Test” page

4.3.2.7 Taking Test

Test 00:01:21

1. Từ nào sau đây không cùng nhóm với các từ còn lại?

Chăm chỉ
 Siêng năng
 Chuyên cần
 Ngoan ngoãn

2. Từ nào là tính từ?

Cuộc sống
 Vui vẻ
 Giúp đỡ
 Thôn xóm

3. Từ nào sau đây không phải là danh từ?

Cuộc sống
 Thành phố
 Đất nước
 Thập thoảng

Check Answer

Figure 9. “Taking Test” page

Field name	Require	Mandatory	Min length	Max length	Description
Answer	Click	Y			Radio button for choosing the correct answer
Check Answer	Click	Y			Button for submitting the test and redirecting to “Test Result” page

Table 8. Components of “Taking Test” page

4.3.2.8 Test Result

Test

Mark: 5/5

1. Từ "ai" trong câu nào dưới đây là từ nghi vấn?

Ông ta gắng hỏi mãi nhưng không ai trả lời

Anh ta đem hoa này tặng ai vậy?

Anh về lúc nào mà không báo cho ai biết cả vây?

Cà xóm này ai mà không biết chủ bé làm lính đó!

Report

Points: 1

2. Với 5 tiếng cho sẵn: kính, yêu, quý, thương, mến, em có thể ghép được bao nhiêu từ ghép có 2 tiếng?

10

7

9

8

Report

Points: 1

3. Từ không đồng nghĩa với từ "hoà bình" là:

bình yên

thanh bình

hiến hoà

cả a, b, c

Report

Points: 1

4. Từ "Bao la" thuộc thể loại từ gì?

động từ

tính từ

danh từ

Report

Points: 1

5. Từ nào là đồng từ?

Cuộc đấu tranh

Ăn cơm

Vui tươi

Niềm vui

Report

Points: 1

Figure 10. “Test Result” page

Field name	Type	Mandatory	Min length	Max length	Description
Report	Button	Y			Button for submitting report about questions that user subjectively consider wrong content.

Table 9. Components of “Test Result” page

4.3.3 Logged-in user

4.3.3.1 User Profile

Profile: veazyus

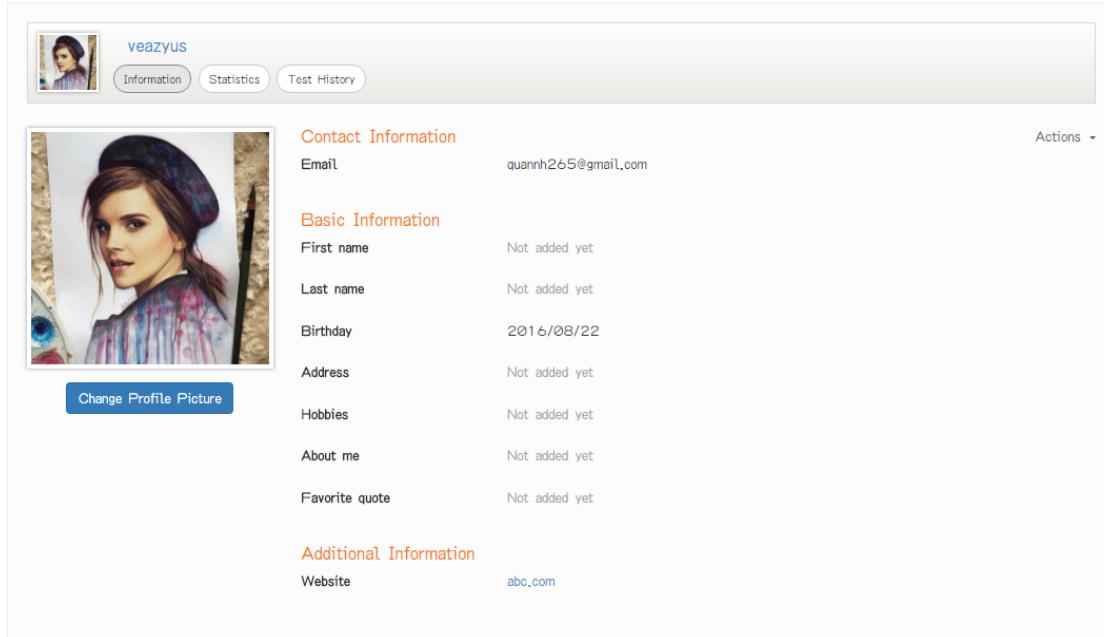


Figure 11. “User Profile” page

Field name	Type	Mandatory	Min length	Max length	Description
Action	Dropdown	N			Dropdown that displays selectable options for user include editing information and changing current password
Information	Button	N			Button for displaying tab for viewing user information

Change Avatar Picture	Button	N			Button for displaying dialog for changing current avatar picture
Statistic	Button	N			Button for displaying tab for viewing user statistics
Test History	Button	N			Button for displaying tab for test history user statistics

Table 10. Components of “User Profile” page

4.3.3.2 User Statistic

Profile: veazys

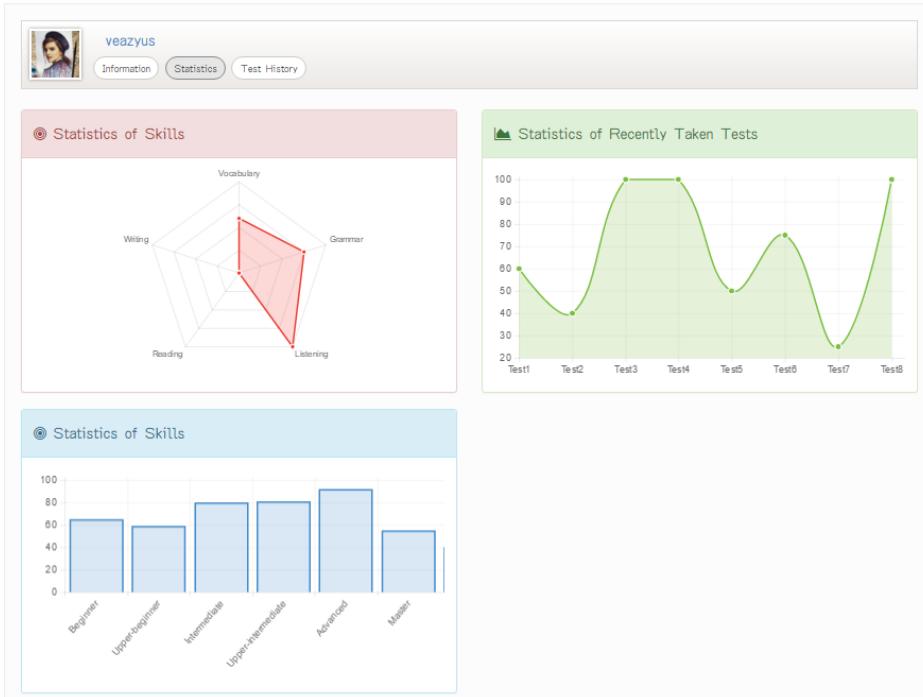


Figure 12. “User Statistic” page

Field name	Type	Mandatory	Min length	Max length	Description
Information	Button	N			Button for displaying tab for viewing user information
Statistic	Button	N			Button for displaying tab for viewing user statistics
Test History	Button	N			Button for displaying tab for test history user statistics

Table 11. Components of “User Statistic” page

4.3.3.3 Test History

Profile: veazyus

The screenshot shows the 'Test History' page with three test records listed:

- Vocabulary Test** (Level: Upper-beginner)

Time allowed	03:00
Time taken	02:44
Result	75/100
- Vocabulary Test** (Level: Upper-beginner)

Time allowed	03:00
Time taken	02:52
Result	25/100
- Listening Test** (Level: Master)

Time allowed	03:30
Time taken	01:46
Result	100/100

At the bottom center of the page is a navigation bar with buttons: < (left), < (left), 1, 2 (highlighted in blue), > (right), > (right), and 3.

Figure 13. “Test History” page

Field name	Type	Mandatory	Min length	Max length	Description
Information	Button	N			Button for displaying tab for viewing user information
Statistic	Button	N			Button for displaying tab for viewing user statistics

Test History	Button	N			Button for displaying tab for test history user statistics
Actions	Dropdown	N			Dropdown that displays selectable options for user include seeing result and retaking test
First	Button	N			Button for displaying the first 5 taken tests in table
Previous	Button	N			Button for navigating to 5 previous taken tests in table
Next	Button	N			Button for navigating to 5 next taken tests in table
Last	Button	N			Button for navigating to 5 last taken tests in table

Table 12. Components of “Test History” page

4.3.3.4 See Result

Test

Result: 75%

1. Tiếng "dòng" nào sau đây khác nghĩa với các tiếng "dòng" còn lại?

Bảng tóm
 Công dòng
 Cảnh dòng
 Bảng chí

2. Từ nào là động từ?

Cuộc đấu tranh
 Ăn cơm
 Vui tươi
 Niềm thương

3. Nghĩa gốc của từ "Ăn"?

Ăn cười
 Ăn cơm
 Ăn ảnh
 Da ăn nắng

4. Từ trái nghĩa với "Hạnh phúc" là gì?

Phúc đức
 Cô đơn
 Bất hạnh
 Qúc khố

Test Preparation

Level: Beginner

Test Skill: Vocabulary

Number of Questions: 5

Create

Figure 14. “See Result” page

Field name	Type	Mandatory	Min length	Max length	Description
Level	Select box	Y			Select box for selecting difficulty level of test
Test Skill	Select box	Y			Select box for selecting skill to be tested
Question Limit	Number	Y			Textbox for entering the number of questions
Create	Button	Y			Button for submitting test setup form and

					redirecting to “Take Test” page
--	--	--	--	--	---------------------------------

Table 13. Components of “See Result” page

4.3.4 Content creator

4.3.4.1 Content Creator Dashboard

The screenshot shows the Veazy Content Creator Dashboard. On the left is a sidebar with a user icon, 'Welcome, siu265', and navigation links: 'Report', 'Lessons', 'Questions', 'Change Language', and 'Logout'. The main area is titled 'Report List' and contains two entries:

- "On Aug 23, 2016, user 2 reported about lesson #2: 'Nội dung câu hỏi sai'. Mark as checked
- "On Aug 23, 2016, user 2 reported about lesson #3: 'Báo án câu hỏi trong phần Practice không chính xác'. Mark as checked

At the bottom of the report list are navigation buttons: '<', '< <', '1', '>', and '> >'.

Figure 15. “Content Creator Dashboard” page

Field name	Type	Mandatory	Min length	Max length	Description
Lessons	Hyperlink	N			Hyperlink that opens lesson management menu
Lesson List	Hyperlink	N			Hyperlink that redirects to “Lesson List” page
Add Lesson	Hyperlink	N			Hyperlink that redirects to “Adding Lesson” page
Question	Hyperlink	N			Hyperlink that opens question management menu
Question List	Hyperlink	N			Hyperlink that redirects to “Question List” page

Add Question	Hyperlink	N			Hyperlink that redirects to “Adding Question” page
Reports	Hyperlink	N			Hyperlink that redirects to “Report List” page
Logout	Hyperlink	N			Hyperlink that logs user out of Veazy system
First	Button	N			Button for displaying the first 5 reports in table
Previous	Button	N			Button for navigating to 5 previous reports in table
Next	Button	N			Button for navigating to 5 next reports in table
Last	Button	N			Button for navigating to 5 last taken tests in table
Mark as checked	Button	N			Button for marking checked report
Change Language	Hyperlink	N			Hyperlink that changes language of website

Figure 14. Components of “Content Creator Dashboard”

4.3.4.2 Lesson List

The screenshot shows a web-based application interface for managing lessons. At the top, there is a search bar labeled "Show only lessons where" with a dropdown menu set to "All Levels" and a green "Filter" button. Below this is a table with the following columns: ID, Title, Level, Current Version, and Actions. The table contains two rows:

ID	Title	Level	Current Version	Actions
2	Bài 1: Chào chí	Beginner	1	Edit Delete
3	Bài 1: Ngày mai trời thế nào?	Upper-beginner	1	Edit Delete

At the bottom of the table, there is a navigation bar with icons for back, forward, and search.

Figure 16. “Lesson List” page

Field name	Type	Mandatory	Min length	Max length	Description
Lesson title	Hyperlink	N			Hyperlink that redirects to “Lesson Detail” page
Edit	Button	N			Hyperlink that redirects to “Editing Lesson” page
Delete	Button	N			Button for deleting corresponding lesson
First	Button	N			Button for displaying the first 5 lessons in table
Previous	Button	N			Button for navigating to 5 previous lessons in table
Next	Button	N			Button for navigating to 5 next lessons in table

Last	Button	N			Button for navigating to 5 last lessons in table
Level	Dropdown	N			Dropdown that displays selectable options for user selects level of lesson
Filter	Button	N			Button for filtering level of lesson

Figure 15. Components of “Lesson List”

4.3.4.3 Lesson Detail

The screenshot shows a lesson detail page with the following information:

- Bài 1: Chào chí**
- Current Version**: 1
- Level**: Beginner
- Description**: Chào hỏi - Greeting Hỏi và giới thiệu tên - Ask and answer name Một số đại từ nhân xưng thông dụng - Some popular personal pronouns Bảng chữ cái tiếng Việt và tên gọi của các chữ cái - Vietnamese alphabet and their names Các nguyên âm đơn - Single vowels Phân biệt thanh ngang và thanh huyền - Distinguish ngang tone to huyền tone
- Content**: Vocab (selected), Grammar, Convo, Listening, Practice, Reading
- Vocab Content** (shown in a box):
 - anh : you (male, older than speaker)
 - bà : Mrs., old women)
 - cảm ơn : thank
- Buttons**: Edit (blue), Delete (red)

Figure 17. “Lesson List” page

Field name	Type	Mandatory	Min length	Max length	Description
Vocab	Hyperlink	N			Hyperlink that displays content of Vocabulary section
Grammar	Hyperlink	N			Hyperlink that displays content of Grammar section
Convo	Hyperlink	N			Hyperlink that displays content of Conversation section
Listening	Hyperlink	N			Hyperlink that displays content of Listening section
Practice	Hyperlink	N			Hyperlink that displays content of Practice section

Reading	Hyperlink	N			Hyperlink that displays content of Reading section
Edit	Button	N			Hyperlink that redirects to “Editing Lesson” page
Delete	Button	N			Button for deleting corresponding lesson

Figure 16. Components of “Lesson Detail”

4.3.4.4 Adding Lesson

Add New Lesson

Lesson Title	<input type="text"/>
Level	Beginner
Description	<input type="text"/>
Content	<div style="display: flex; justify-content: space-between;"> Vocab Grammar Convo Listening Practice Reading </div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> A B C D E F G H I J K L M N O P Q R S T U V W X Y Z “ ” - % Image Video File List Clock Text Pencil Eraser Delete Search Help </div>
	<input type="button" value="Create"/>

Figure 18. “Adding Lesson” page

Field name	Type	Mandatory	Min length	Max length	Description
Lesson Title	Text	Y	0		Textbox for entering lesson title
Level	Select box	Y			Select box for choosing difficulty level of lesson
Description	Text area	Y			Text area for entering lesson description
Vocab	Hyperlink	Y			Hyperlink that displays editor of Vocabulary section
	Text area	Y			Text area for entering content of Vocabulary section

Grammar	Hyperlink	Y			Hyperlink that displays editor of Grammar section
	Text area	Y			Text area for entering content of Grammar section
Convo	Hyperlink	Y			Hyperlink that displays editor of Conversation section
	Text area	Y			Text area for entering content of Conversation section
Listening	Hyperlink	Y			Hyperlink that displays editor of Listening section
	Text area	Y			Text area for entering content of Listening section
Practice	Hyperlink	Y			Hyperlink that displays editor of Practice section
	Text area	Y			Text area for entering content of Practice section
Reading	Hyperlink	Y			Hyperlink that displays editor of Reading section

	Text area	Y			Text area for entering content of Reading section
Create	Button	Y			Button for submitting new lesson

Table 17. Components of “Adding Lesson” page

4.3.4.5 Editing Lesson

The screenshot shows the 'Edit Lesson' interface. At the top, there's a title bar labeled 'Edit Lesson'. Below it, there are several input fields:

- Lesson Title:** A text input field containing 'Bài 1: Chào chí'.
- Current Version:** A dropdown menu showing '1'.
- Level:** A dropdown menu set to 'Beginner'.
- Description:** A text area containing 'Chào hỏi - Greeting' and 'Hỏi và giới thiệu tên - Ask and answer name'.
- Content:** A tabbed section with tabs for 'Vocab', 'Grammar', 'Convo', 'Listening', 'Practice', and 'Reading'. The 'Reading' tab is selected. Below the tabs is a rich text editor toolbar with various icons for bold, italic, underline, etc. Underneath the toolbar is a preview area showing sample text: 'anh : you (male, older than speaker)', 'bà : Mrs., old women)', and 'cám ơn : thank'. A character count of '71' is shown in the bottom right corner of the preview area.

At the bottom of the form, there are three buttons: 'Update' (green), 'Save draft' (blue), and 'Back to Detail'.

Figure 19. “Editing Lesson” page

Field name	Type	Mandatory	Min length	Max length	Description
Lesson Title	Text	Y			Textbox for entering lesson title
Level	Select box	Y			Select box for choosing difficulty level of lesson
Description	Text area	Y			Text area for entering lesson description

Vocab	Hyperlink	Y			Hyperlink that displays editor of Vocabulary section
	Text area	Y			Text area for entering content of Vocabulary section
Grammar	Hyperlink	Y			Hyperlink that displays editor of Grammar section
	Text area	Y			Text area for entering content of Grammar section
Convo	Hyperlink	Y			Hyperlink that displays editor of Conversation section
	Text area	Y			Text area for entering content of Conversation section
Listening	Hyperlink	Y			Hyperlink that displays editor of Listening section
	Text area	Y			Text area for entering content of Listening section
Practice	Hyperlink	Y			Hyperlink that displays editor of Practice section

	Text area	Y			Text area for entering content of Practice section
Reading	Hyperlink	Y			Hyperlink that displays editor of Reading section
	Text area	Y			Text area for entering content of Reading section
Create	Button	Y			Button for submitting new lesson
Save Draft	Button	Y			Button for saving lesson
Back to Detail	Button	Y			Button for backing to detail of lesson page

Table 18. Components of “Editing Lesson” page

4.3.4.6 Question List

The screenshot shows a user interface for managing questions. At the top, there is a search bar labeled "Show only questions where" with dropdown menus for "Type" (All Types), "Level" (All Levels), and "Skill" (All Skills). Below these is a green "Filter" button. The main area contains a table with columns: ID, Content, Type, Level, Skill, and Actions. The table has 6 rows, each representing a question with ID 2 through 7. All questions are of type "Single", level "Beginner", and skill "Vocabulary". Each row has "Edit" and "Delete" buttons in the "Actions" column. At the bottom of the table is a navigation bar with icons for back, forward, and search.

ID	Content	Type	Level	Skill	Actions
2	Click here to see content	Single	Beginner	Vocabulary	Edit Delete
3	Click here to see content	Single	Beginner	Vocabulary	Edit Delete
4	Click here to see content	Single	Beginner	Vocabulary	Edit Delete
5	Click here to see content	Single	Beginner	Vocabulary	Edit Delete
6	Click here to see content	Single	Beginner	Vocabulary	Edit Delete

Figure 20. “Question List” page

Field name	Type	Mandatory	Min length	Max length	Description
Click here to see content	Hyperlink	N			Hyperlink that redirects to “Question Detail” page
Edit	Button	N			Hyperlink that redirects to “Editing Question” page
Delete	Button	N			Button for deleting corresponding question
First	Button	N			Button for displaying the first 5 questions in table

Previous	Button	N			Button for navigating to 5 previous questions in table
Next	Button	N			Button for navigating to 5 next records in table
Last	Button	N			Button for navigating to 5 last questions in table
Type	Dropdown	N			Dropdown that displays selectable options for user selects type of question
Level	Dropdown	N			Dropdown that displays selectable options for user selects level of question
Skill	Dropdown	N			Dropdown that displays selectable options for user selects skill of question
Filter	Button	N			Button for filtering question

Table 19. Components of “Question List page**4.3.4.7 Question Detail**

Question #2

Question Type	Single
Level	Beginner
Test Skill	Vocabulary
Time allowed	00:30 (mm:ss)
File Attachment	Some file

Question

Từ nào sau đây viết sai chính tả?

Answers

- Gà ghè
- Ngượng ngùu
- Kẽm capse
- Kim cương

[Edit](#) [Delete](#)

Figure 21. “Question Detail” page (single question)

Question #15

Question Type	Group
Level	Beginner
Test Skill	Reading
Time allowed	05:30 (mm:ss)
File Attachment	Some file

Content

Click to hide shared content ^
 Rung cây im lặng qua, Một tiếng lả rời lúc này cũng có thể khiến người ta giật mình, La què, chìm chốc chẳng nghe con nào kêu, Hay vua có tiếng chim ở một nơi nào xa lánh, vì không chú ý mà tôi không nghe chừng.Giờ bắt đầu nói rõ ràò với khói mặt trời đang tưởn sáng vàng rực xung quanh đất, Một lần hót đất nhẹ nhẹ tỏa tên, phả mồ hôi những cây cối đó, rồi tan biến theo hơi ấm mặt trời, Phút yên tĩnh của riêng ban mai dần biến đi,Chim hót lúi lo, Nắng bốc hương hoa tràm thêm ngày ngắn, Gió đưa mùi hương ngọt lan xa, phảng phất khẩu nồng, Mây con kỳ nhông nắm phơi lông trên gác cây mạc, sắc da lung lung biến đổi từ xanh hóa vàng, từ vàng hóa đỏ, từ đỏ hóa tím xanh...Con bướm đong dây cánh mới, ròn ròn bò túi, Nghệ tiếng chân con chو săn ngay hilam, những con vật thuộc loài bò sát có bốn chân to hòn ngón chân cái kia liền quết chiếc đuôi dài chạy từ tần, con não chỗ gốc cây thi biến thành màu xám và cây, con deo trên tần là ngồi thi biến ra màu xanh ta ngát,

Question 1

Những chi tiết miêu tả cảnh yên tĩnh của vùng phương Nam là?

- Tiếng chim hót từ xa vọng lại
- Chìm chốc chẳng con nào kêu, một tiếng lả rời cũng khiến người ta giật mình
- Gió bắt đầu nổi tên

Question 2

Mùi hương của hoa tràm như thế nào?

- Nhè nhẹ tỏa tên
- Tan dần theo hơi ấm mặt trời
- Thơm ngày ngắn, phảng phất khẩu nồng

[Edit](#) [Delete](#)

Figure 22. “Question Detail” page (group question)

Field name	Type	Mandatory	Min length	Max length	Description
Click to show shared content	Hyperlink				Hyperlink that displays shared content of group question

Edit	Button	N			Button that redirects to “Editing Question” page
Delete	Button	N			Button for deleting question

Table 20. Components of “Question Detail” page

4.3.4.8 Adding Question

The screenshot shows a form titled "Add Question". It includes the following fields:

- Question Type: Single
- Level: Beginner
- Test Skill: Listening
- Question: Here comes the content of question
- Answers:
 - Here comes the content of answer #1
 - Here comes the content of answer #2
- File Attachment: A file input field with a "Browse ..." button.
- Time allowed: 00:30

At the bottom is a green "Submit" button.

Figure 23. “Adding Question” page (single question)

Field name	Type	Mandatory	Min length	Max length	Description
Question type	Select box	Y			Select box for choosing type of question
Level	Select box	Y			Select box for choosing difficulty level of question

Test Skill	Select box	Y			Select box for choosing skill to be tested of question
Question	Text	Y			Textbox for entering question
Check box	Check box	Y			Checkbox for marking which answer is true
Answer	Text	Y			Textbox for entering answer content
X	Button	N			Button for removing an answer
+	Button	N			Button for adding one more answer to the question
Browser	File	N			Button for display dialog of selection audio
Submit	Button	Y			Button for submitting new question
Time Allowed	Textbox	N			Textbox for displaying time of question
Clock	Icon	N			Textbox for setting time of question

Table 21. Components of “Adding Question” page (single question)

Add Question

Question Type	<input type="text" value="Group"/>
Level	<input type="text" value="Beginner"/>
Test Skill	<input type="text" value="Vocabulary"/>
Content	<div style="border: 1px solid black; padding: 5px;"><p>Here comes the content of question #1</p><p><input type="checkbox"/> Here comes the content of answer #1 x</p><p><input type="checkbox"/> Here comes the content of answer #2 x</p><p style="text-align: center;">+</p></div>
Question 1	<div style="border: 1px solid black; padding: 5px;"><p>Here comes the content of question #2</p><p><input type="checkbox"/> Here comes the content of answer #1 x</p><p><input type="checkbox"/> Here comes the content of answer #2 x</p><p style="text-align: center;">+</p></div>
Question 2	<div style="border: 1px solid black; padding: 5px;"><p>00:30</p><p style="text-align: right;">Submit</p></div>

Figure 24. “Adding Question” page (group question)

Field name	Type	Mandatory	Min length	Max length	Description
Question type	Select box	Y			Select box for choosing type of question
Level	Select box	Y			Select box for choosing difficulty level of question
Test Skill	Select box	Y			Select box for choosing skill to be tested of question
Content	Text area	Y			Text area for entering shared content of group question

Question	Textbox	Y			Textbox for entering content of question in group
X	Button	N			Button for removing question in group
Check box	Check box	Y			Checkbox that marks the correct answer in question
Answer	Textbox	Y			Textbox for entering content of answer
X	Button	Y			Button for removing answer in question
+	Button	N			Button for adding one more answer
+	Button	N			Button for adding one more question
Submit	Button	Y			Button for submitting new question
Time Allowed	Textbox	Y			Textbox for setting time of question

Table 22. Components of “Adding Question” page (group question)

4.3.4.9 Editing Question

The screenshot shows a form titled 'Edit question #3'. The fields are as follows:

- Question Type:** Single (selected)
- Level:** Beginner
- Test Skill:** Vocabulary
- Question:** Từ nào sau đây không cùng nhóm với các từ còn lại?
- Answers:**
 - Châm chí (X)
 - Siêng năng (X)
 - Chuyên cần (X)
 - Ngoan ngoãn (X)
- Time allowed:** 00:30

At the bottom left are 'Update' and 'Back to Detail' buttons.

Figure 25. “Editing Question” page (single question) (1)

Field name	Type	Mandatory	Min length	Max length	Description
Question type	Select box	Y			Select box for choosing type of question
Level	Select box	Y			Select box for choosing difficulty level of question
Test Skill	Select box	Y			Select box for choosing skill to be tested of question
Question	Textbox	Y			Textbox for entering content of question in group

Check box	Check box	Y			Checkbox that marks the correct answer in question
Answer	Textbox	Y			Textbox for entering content of answer
+	Button	N			Button for adding more selectable answers of the question
X	Button	N			Button for deleting answer of question
Submit	Button	Y			Button for submitting the question content
Back to Detail	Button	Y			Button for backing to Question Detail page
Time Allowed	Textbox	N			Textbox for displaying time of question
Clock	Icon	N			Textbox for setting time of question

Table 23. Components of “Editing Question” page (single question) (1)

The screenshot shows the 'Edit question #15' interface. At the top, there are dropdown menus for 'Question Type' (Group), 'Level' (Beginner), and 'Test Skill' (Reading). Below these is a rich text editor toolbar with various icons for bold, italic, underline, superscript, and other text formats. The main content area contains a paragraph of Vietnamese text about a tree and its bark. A small number '969' is visible in the bottom right corner of the content area.

Figure 26. “Editing Question” page (group question) (2)

Question 1
Những chi tiết miêu tả cảnh yên tĩnh của vùng phương Nam là?

Tiếng chim hót từ xa vọng lại ×

Chim chóc chẳng con nào kêu, một tiếng lá rơi cũng khiến người ta giật mình ×

Gió bắt đầu nổi lên ×

Question 2
Mùi hương của hoa tràm như thế nào?

Nhè nhẹ tỏa lên ×

Tan dần theo hơi ấm mặt trời ×

Thơm ngay ngắn, phảng phất khắp nungan ×

Time allowed: 05:30

[Update](#) [Back to Detail](#)

Figure 27. “Editing Question” page (group question) (3)

Field name	Type	Mandatory	Min length	Max length	Description
Question type	Select box	Y			Select box for choosing type of question
Level	Select box	Y			Select box for choosing difficulty level of question
Test Skill	Select box	Y			Select box for choosing skill to be tested of question
Content	Text	Y			Textarea for entering the shared content used by all questions
Question	Text	Y			Textarea for entering content of a single question in group

X	Button	N			Button for deleting the corresponding question in group
Check box	Check box	Y			Checkbox that marks the correct answer in question
Answer	Text	Y			Textbox for entering content of answer
X	Button	Y			Button for deleting the corresponding answer
+	Button	N			Button for adding more question to the group
+	Button	N			Button for adding more answer to the question
Submit	Button	Y			Button for submitting the group question
Back to Detail	Button	Y			Button for backing to Question Detail page
Time Allowed	Textbox	N			Textbox for displaying time of question
Clock	Icon	N			Textbox for setting time of question

Table 24. Components of “Editing Question” page (group question)(2)

4.3.5 Admin

4.3.5.1 Admin Dashboard

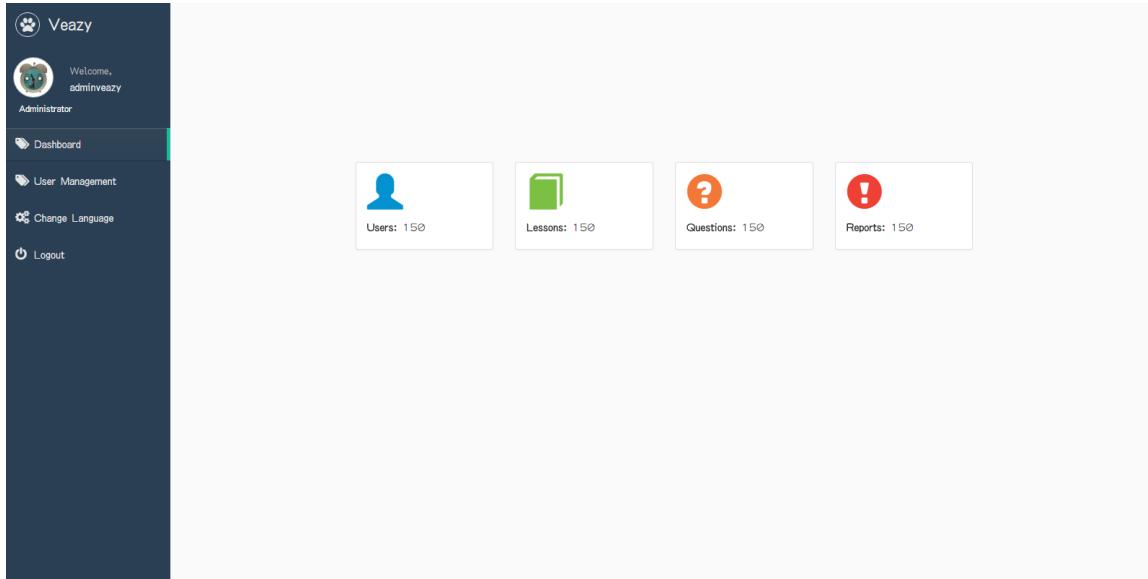


Figure 28. “Admin Dashboard” page

Field name	Type	Mandatory	Min length	Max length	Description
Dashboard	Hyperlink	N			Hyperlink that redirects to “Admin Dashboard” page
User Management	Hyperlink	N			Hyperlink that redirects to “User management” page
Change Language	Hyperlink	N			Hyperlink that changes language of website
Logout	Hyperlink	N			Hyperlink that logs user out of Veazy system

Table 25. Components of “Admin Dashboard” page

4.3.5.2 Member List

The screenshot shows a web-based application interface for managing users. At the top left, there is a search/filter section titled "Show only users where". It contains three dropdown menus: "Role" (with options like "User", "Content Editor", "Administrator"), "Status" (with options like "Active", "Banned", "Suspended"), and "Username" (with a placeholder "Search..."). Below these is a blue "Filter" button. To the right of this is a table listing three users:

Username	Status	Role	Actions
Crimson	Active	User	Manage
editor	Active	Content Editor	Manage
anonymous	Banned	User	Manage

At the bottom center of the page is a navigation bar with icons for back, forward, and search.

Figure 29. “Member List” page

Field name	Type	Mandatory	Min length	Max length	Description
Role	Dropdown	N			Dropdown that displays selectable options for user to selects role of member
Status	Dropdown	N			Dropdown that displays selectable options for user to selects status of member
Username	Textbox	N			Textbox for entering username of member

Filter	Button	N			Button for filtering member by role, status or username
Username	Hyperlink	N			Hyperlink that shows “Member Information Detail” popup
Manage	Dropdown	N			Dropdown that displays selectable options for user include bans member, assigns as editor and assigns as admin
First	Button	N			Button for displaying the first 5 members in table
Previous	Button	N			Button for navigating to 5 previous members in table
Next	Button	N			Button for navigating to 5 next members in table
Last	Button	N			Button for navigating to 5 last members in table

Table 26. Components of “Member List” page